aerospace

MDPI

Review

The Evolving Paradigm of Reliability Engineering for Complex Systems: A Review from an Uncertainty Control Perspective

Zhaoyang Zeng, Cong Lin *, Wensheng Peng and Ming Xu

China Aero-Polytechnology Establishment, Beijing 100028, China; zengzy003@avic.com (Z.Z.); pengws002@avic.com (W.P.); xum058@avic.com (M.X.)
* Correspondence: linc002@avic.com

Abstract

Traditional reliability engineering paradigms, originally designed to prevent physical component failures, are facing a fundamental crisis when applied to today's software-intensive and autonomous systems. In domains like aerospace, critical risks no longer stem solely from the aleatory uncertainty of hardware breakdowns, but increasingly from the deep epistemic uncertainty inherent in complex systematic interactions and non-deterministic algorithms. This paper reviews the historical evolution of reliability engineering, tracing the progression through the Statistical, Physics-of-Failure, and Prognostics Eras. It argues that while these failure-centric frameworks perfected the management of predictable risks, they are structurally inadequate for the "unknown unknowns" of modern complexity. To address this methodological vacuum, this study advocates for an imperative shift towards a fourth paradigm: the Resilience Era. Grounded in the principles of Safety-II, this approach redefines the engineering objective from simply minimizing failure rates to ensuring mission success and functional endurance under uncertainty. The paper introduces uncertainty control (UC) as the strategic successor to uncertainty quantification (UQ), proposing that safety must be architected through behavioral constraints rather than prediction alone. Finally, the paper proposes a new professional identity for the practitioner: the system resilience architect, tasked with designing adaptive architectures that ensure safety in an era of incomplete knowledge.

Keywords: reliability engineering; uncertainty quantification; resilience engineering; uncertainty control

1. Introduction

1.1. The Growth of Complexity in Safety-Critical Systems

The landscape of modern engineering is being reshaped by a profound and accelerating increase in system complexity. This is most evident in safety-critical domains such as aerospace, automotive, and maritime industries, where a fundamental transition is underway from systems defined by their physical and mechanical properties to those defined by their software, connectivity, and increasingly, their autonomy [1,2]. This evolution is not merely a linear extrapolation of past trends but represents a step-change in the nature of the systems that engineers must design, analyze, and certify.

A prime example of the system with the above-mentioned new properties is the emergence of electric Vertical Take-Off and Landing (eVTOL) aircraft for Urban Air Mobility (UAM). Unlike traditional aircraft, eVTOLs are characterized by highly integrated distributed electric propulsion (DEP) systems, where aerodynamic forces, structural loads,

Check for updates

Academic Editor: Sujin Bureerat

Received: 30 December 2025
Revised: 9 February 2026
Accepted: 10 February 2026
Published: 13 February 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

and flight control logic are deeply and non-linearly coupled [3]. To achieve the necessary performance, stability, safety and reliability in dynamic urban environments, these vehicles rely on advanced, often non-deterministic control algorithms, such as Model Predictive Control (MPC) or even machine learning (ML)-based controllers [4,5]. This trend also exists in modern civil aircraft whose avionics system has been upgraded to Integrated Modular Avionics (IMA) architectures. IMA transforms the aircraft into a software-defined platform, where multiple functions of differing criticalities share a common set of computational resources, making system safety contingent upon the correct and non-interfering interaction of countless software partitions [6,7]. Likewise, the progression from static, gain-scheduled flight controllers to adaptive flight control systems introduces a new level of performance alongside significant challenges in verification and predictability [8,9].

The distinctive feature of these modern systems is their high degree of integration and interactivity. They are no longer loose collections of independent subsystems, but are tightly coupled architectures where a state change or a fault in one domain can propagate instantaneously and non-linearly across the entire platform, leading to system-level hazards that are impossible to foresee through traditional decomposition analysis [10,11]. Furthermore, the introduction of autonomy fundamentally alters the human–machine relationship. Operators are moving from direct manual control to a supervisory role, creating new classes of risks related to mode confusion, loss of situational awareness, and the opacity of autonomous decision-making processes [12,13].

Consequently, the defining characteristic of these modern systems is not just their scale, but their emergent capability. They are strictly "beyond the sum of their parts", a phrase we use throughout this paper to describe systems where safety is determined by non-linear software–hardware interactions rather than the aggregate reliability of individual physical components. As these systems become ubiquitous, the engineering community faces a critical challenge: the existing reliability and safety assurance frameworks, developed for a simpler, more deterministic era, are proven to be inadequate for managing the risks posed by those uncertainty factors.

1.2. The Emerging Crisis of Traditional Reliability Paradigms

The historical paradigms of reliability engineering, including statistical methods, physics modeling and prognostic-based approach, share a unifying philosophical foundation: they are fundamentally failure-centric. This worldview posits that system safety is primarily a function of component reliability. The core assumption is that accidents and system failures are the result of a chain of cascading or concurrent component faults—a broken part, a software bug, or a sensor that stops working [14]. Consequently, the primary engineering objective has been to understand, predict, and prevent these component failures. This philosophy is formally recognized in safety science as "Safety-I," which defines safety as the absence of accidents and incidents [15].

This failure-centric perspective is deeply embedded in the classical tools of reliability and safety evaluation. Techniques like Failure Modes and Effects Analysis (FMEA) and Fault Tree Analysis (FTA) are built upon a reductionist premise: that by exhaustively analyzing the ways individual components can fail and tracing their consequences, one can understand and mitigate system-level risk [16]. The underlying logic is that if all components are made sufficiently reliable, then the whole system will be safe. This can be conceptually represented as

$$\text{System Safety} = 1 - P\{\text{System Accident}\} = 1 - P\{\cup\text{Component failure}\} \quad (1)$$

where $\cup$ represents the union of all possible sets. This model, which assumes that system accidents can be decomposed into a sum or combination of component failures, has been

remarkably successful for decades in improving the safety of mechanical and electromechanical systems. However, this fundamental assumption is now in direct conflict with the reality of modern complex systems.

The main crisis facing traditional reliability is the increasing prevalence of accidents in which no component has failed according to its individual failure criteria. In highly integrated or software-intensive systems, catastrophic failures are increasingly caused not by broken parts, but by unsafe interactions among components that are all functioning exactly as designed [10]. These are known as emergent failure, i.e., system-level properties, that cannot be predicted by analyzing components in isolation. The accidents involving the Boeing 737 MAX Maneuvering Characteristics Augmentation System (MCAS) serve as a tragic example. According to the National Transportation Safety Board (NTSB) investigation, the individual components were operating as specified, but their collective behavior, orchestrated by software and driven by unexpected environmental inputs, created a hazardous system state [17]. This demonstrates that in modern systems, system failure has shifted from physical components to intellectual design involving the requirements, the control logic, and the assumptions about the operational environment. The comparison of traditional and systemic accident causality is listed in Table 1.

Table 1. The comparison of traditional and systemic accident causality.

Attribute	Component Failure Model	Systemic Accident Model
Locus of Cause	Physical or software component failure	Unsafe interactions between non-failed components
Causal Model	Linear chain of events	Complex feedback loops and systemic structure
Safety View	"Safety-I": Safety is the absence of failures	"Safety-II": Safety is an emergent system property
Assumption	Reliable components lead to a safe system	A safe system successfully controls its behavior

The rise in interactional accidents has created a methodological crisis for traditional reliability engineering. A dangerous gap now exists between the systemic nature of safety in modern platforms and the component-focused tools used to evaluate it. For example, FMEA, by focusing on the effects of individual component failures, is structurally incapable of identifying hazards that arise from unsafe interactions between multiple, non-failed components. Although FTA can model combinations of events, it struggles when the root events are not component failures but flawed requirements, complex software logic, or incorrect assumptions about human behavior [18]. The crisis, therefore, is a methodological vacuum between the complex systems we are building and the methods we use for guaranteeing their safety. This necessitates a redefinition of the paradigm of reliability engineering, shifting the focus from preventing component failures to controlling the systemic behaviors and uncertainties that lead to hazardous states.

The rising significance of epistemic uncertainty and the blind spots it creates for traditional methods are evidenced by recent high-profile failures where systems operated strictly within their specifications yet caused catastrophes.

1. The Boeing 737 MAX

The MCAS accidents were not caused by the stochastic failure of the flight computer, but by a flaw in the system's logic design. The software trusted a single sensor input without adequate cross-checking, a vulnerability arising from the designers' incomplete knowledge of how the system would interact with pilots under high workload. The JATR report highlighted that the design assumptions regarding pilot reaction times were fundamentally flawed [19].

2. The Uber ATG Accident

In the 2018 Tempe crash, the autonomous vehicle's sensors successfully detected the pedestrian 5.6 s prior to impact. However, no braking command was issued because the decision-making algorithm was calibrated to suppress "false positives" to ensure ride smoothness. The accident resulted from a deliberate design choice that misclassified a real threat as noise, demonstrating that compliance with component reliability requirements cannot prevent hazards born from flawed functional logic [20].

3. Phantom Braking Events

Modern ADAS frequently exhibit "phantom braking", where vehicles decelerate dangerously despite clear roads. These incidents stem from AI perception models encountering environmental patterns that lie outside their training distribution. Here, the risk source is purely the AI's lack of knowledge about the open world, rather than any hardware degradation [21].

1.3. The Shifting Nature of Uncertainty: From Aleatory to Epistemic

The crisis facing traditional reliability paradigms is not merely a matter of scale, but a fundamental shift in the very nature of the uncertainty that engineers must confront. For decades, the discipline achieved success by mastering the management of aleatory uncertainty. However, the complex, software-defined modern systems are increasingly dominated by epistemic uncertainty [22].

- Aleatory uncertainty

It refers to the inherent, irreducible randomness or variability in a physical system or its environment. Often termed "stochastic uncertainty" or "variability", it represents the natural fluctuations that persist even with perfect knowledge of the system [23]. Classic examples in aerospace include microscopic variations in material fatigue properties, atmospheric turbulence, and manufacturing tolerances within an acceptable range. The defining characteristic of aleatory uncertainty is that, given sufficient data, it can be accurately described by a probability distribution, allowing its impact to be quantified with statistical confidence.

- Epistemic uncertainty

It refers to uncertainty stemming from a lack of knowledge on the part of the observer or modeler. Often termed "cognitive uncertainty" or "reducible uncertainty," it represents a deficit in our understanding that could, in principle, be reduced by gathering more data, developing better models, or gaining more experience [24].

Hence, the total uncertainty in any prediction about a system's behavior is a combination of both. A simplified conceptual model can be expressed as

$$y = f_{\text{model}}(x_{aleatory}, \theta_{epistemic}) + \delta_{aleatory} + \varepsilon_{epistemic} \quad (2)$$

where y is the true system output, f_{model} is the established imperfect computer model, $x_{aleatory}$ represents inputs with inherent randomness, $\theta_{epistemic}$ represents model parameters we are unsure about, $\delta_{aleatory}$ is the model error and $\varepsilon_{epistemic}$ represents the error of all other noise.

Traditional reliability engineering excelled because its primary focus was on characterizing the aleatory terms $\{x_{aleatory}, \delta_{aleatory}\}$ through extensive testing and statistical analysis by assuming the epistemic terms $\{\theta_{epistemic}, \varepsilon_{epistemic}\}$ were negligible or could be managed. However, this assumption is completely violated for modern complex systems, where epistemic uncertainty is no longer a secondary factor but has emerged as a critical

source of system failure that traditional methods struggle to address. This consequence arises from several interconnected sources as follows:

1. Model uncertainty: As systems like eVTOL operate in novel flight regimes (e.g., transition flight in urban canyons), the physics-based simulation models used for their design become less reliable. The discrepancy between the model and reality grows, representing a significant form of epistemic uncertainty [25].
2. Algorithmic uncertainty: The behavior of advanced control algorithms, especially those based on AI/ML, introduces a new form of epistemic uncertainty. For a deep neural network, we lack the complete "knowledge" to predict its output for every possible input, particularly for out-of-distribution scenarios not seen during training [26,27].
3. Operational uncertainty: For entirely new operational concepts like Urban Air Mobility (UAM), there is no historical data to build probabilistic models of the environment. This "zero-sample" problem—where we lack knowledge of traffic densities, weather patterns in urban microclimates, or novel human–machine interaction failure modes—is a pure form of epistemic uncertainty [28].

This fundamental shifting nature of uncertainty is not merely an academic observation. It is being actively addressed and codified within the aerospace industry's most critical safety standards. The evolution from SAE ARP4754A to ARP4754B provides direct evidence of this change. The standard's deliberate replacement of the term "unintended function" with "unintended behavior" is a landmark philosophical revolution. An "unintended function" implies a discrete, solvable design error. In contrast, an "unintended behavior" is defined as an "unexpected operation of integrated aircraft systems" that can arise even when all components are functioning as specified [29]. This acknowledges that safety is no longer just about the reliability of individual parts but is an emergent property of the system's interactions—a problem deeply rooted in epistemic uncertainty about those interactions.

To address this new reality, SAE ARP4761A places greater emphasis on specific analytical techniques designed to identify systemic and interactional risks that FMEA/FTA might miss [30]. Key methods that are now central to the safety process for complex systems include: Cascading Effects Analysis (CEA), Common Cause Analysis (CCA) and Investigation of Unintended Behaviors. These methods tell an industry-wide acknowledgment that the primary threat is no longer just the predictable randomness of component failures, but the epistemic uncertainty surrounding the emergent behavior of the system.

1.4. The Necessity of Change in Uncertainty Management Process

The uncertainty management process consists of uncertainty identification, uncertainty quantification and uncertainty control. For traditional systems where risk was primarily driven by the aleatory uncertainty of component failures, the key task was uncertainty quantification, as the influencing factors were easy to identify (e.g., material fatigue or electronic part failure) and their relationships could be modeled (e.g., via fault trees or stochastic processes). The uncertainty quantification helps engineers to determine the effects created by those factors, given sufficient data [31]. The primary output of this paradigm was a calculated risk metric, such as a probability of failure, which informed design decisions.

However, for the modern system, this traditional paradigm reaches its limits, because the factors influencing safety and reliability have become not only more numerous but also qualitatively different. Key influencing factors are now often difficult to identify totally; the coupling relationships between them are non-linear and difficult to model. Consequently, their effects are often impossible to quantify with high confidence [32]. So, we propose

that when a system's behavior under uncertainty can no longer be precisely identified and quantified a priori, the only viable strategy is to control its behavior to stay on the right track during operation [33].

This review argues that to ensure the safety of complex systems, the reliability engineering paradigm must undergo an imperative shift: from a philosophy focused on the passive assessment of uncertainty quantification (UQ) to one centered on the active practice of uncertainty control (UC). Unlike the traditional approach, which primarily aims to calculate risk metrics (e.g., failure rates) based on static assumptions, the UC paradigm seeks to architecturally constrain system behaviors within a valid safety envelope. It is crucial to clarify that this shift does not equate to abandoning design-time analysis for run-time adaptation. Instead, UC encompasses both phases. In the design phase, methodologies like STPA are used to identify hazardous control actions and define safety constraints. In the operational phase, architectures like run-time assurance (RTA) dynamically enforce these constraints. Thus, the distinction of UQ and UC is not between "design" and "operation", but between "predicting failure probability" and "enforcing safety boundaries" regardless of the uncertainty source.

The discipline of reliability engineering has not been static; rather, it has undergone an accretive evolution, with each new paradigm building upon the last to address increasingly complex challenges. The first three major eras—Statistical, Physics-of-Failure, and Prognostics—represent a multi-decade effort to master the risks associated with component and subsystem failures, primarily driven by aleatory uncertainty. This section will trace this historical layering, detailing how each stage added a new level of proactive capability while retaining the essential tools of the past. By doing so, it will build a compelling argument for why this mature, component-focused philosophy, while still necessary, is no longer sufficient. It has reached its conceptual boundary, creating the imperative for an emerging fourth paradigm—Resilience—which envelops the previous stages to address the systemic, interaction-driven uncertainties for which they were not designed. The main characteristics of the paradigm in each era are listed in Table 2.

Table 2. The characteristics of the reliability paradigm in each era.

Characteristics	Statistical Paradigm	Physic-of-Failure Paradigm	Prognostic Paradigm	Resilience Paradigm
Developed era	1950s–1970s	1980s–1990s	2000s–2010s	2020s–present
Focus	macro-level failure data	causal failure mechanisms	real-time component health	systemic behavior
Goal	quantify population reliability	proactive failure prevention	predict impending failures	mission success under uncertainty
Methodology	life data analysis	FMEA/FTA, degradation models	PHM, CBM, HUMS, CMS	RTA, STPA, Resilience Engineering
Approach	reactive	preventive	predictive	adaptive

1.5. The Contributions and Innovations

While reliability engineering is a well-researched field, the existing literature reviews typically operate within specific methodological areas. For instance, extensive surveys exist for statistical reliability models [34], Physics-of-Failure mechanisms in electronics [35], or deep learning algorithms for Prognostics and Health Management (PHM) [36]. Few studies have attempted to bridge the gap between the component-centric techniques and the system-centric safety challenges of complex systems. Therefore, the main contributions and innovations of this paper can be concluded as the following three aspects:

1. We provide a unified evolutionary narrative based on uncertainty

This study connects the four eras through a unified theoretical view, i.e., the shift from managing aleatory uncertainty to battling epistemic uncertainty. This perspective explains the fundamental necessity of the paradigm shift, rather than just describing how it happened.

2. We propose the synthesis of the uncertainty control (UC) Framework

We put forward the UC paradigm not as a mere aggregation of existing tools, but as a transdisciplinary synthesis. As summarized in Table 1, the proposed framework integrates Control Theory, Safety Engineering and Artificial Intelligence. This synthesis innovates by redefining safety from a probability estimation problem to a control architecture problem.

3. We invent architectural guidance to enhance system resilience

Beyond theoretical discussion, this research delineates the specific methodological path towards system resilience. By coupling STPA (for design-time analysis) with run-time assurance (RTA) architectures, we provide a concrete engineering blueprint for handling the epistemic uncertainty in highly integrated complex systems, e.g., aerospace systems, filling the blank left by the traditional failure-centric paradigm.

1.6. The Organization of This Article

The remainder of this paper is structured to trace the evolutionary trajectory of reliability engineering through four distinct paradigms. Section 2 reviews the Statistical Era, focusing on empirical modeling of component failures under aleatory uncertainty using probability theory. Section 3 discusses the Physics-of-Failure Era, which shifts the focus to "white-box" causal mechanisms to design reliability into physical components. Section 4 examines the Prognostics Era, highlighting the transition toward dynamic, real-time health management and Remaining Useful Life (RUL) prediction via data-driven and hybrid approaches. Section 5 establishes the core argument for the Resilience Era, proposing a strategic shift from uncertainty quantification (UQ) to uncertainty control (UC). It details the philosophy of operating under deep epistemic uncertainty and introduces key methodologies—Systems-Theoretic Process Analysis (STPA) and run-time assurance (RTA)—as the architectural pillars for ensuring mission success. Finally, Section 6 summarizes the study and outlines the future outlook for this discipline.

2. The Statistical Era: Reliability as an Empirical Science

2.1. Core Philosophy: Treating Failure as a Black-Box Stochastic Process

Reliability engineering as a formal discipline emerged in the 1950s, primarily driven by the urgent need to address the unacceptably high failure rates of increasingly complex military electronics and aerospace systems [37]. The foundational paradigm of this period was rooted in probability theory and mathematical statistics, treating the complex system as an analytical "black box". The core philosophy was that failures, regardless of their intricate physical origins, could be modeled as stochastic events occurring over time. The primary objective was not to understand the root causes of failure but to empirically characterize the failure behavior of a large population of components based on extensive test or operational field data.

This approach was a direct and practical response to the dominant challenge of that time: aleatory uncertainty, the inherent and irreducible randomness in material properties, manufacturing processes, and operational loads [23]. By assuming failures were random variables, engineers could use statistical methods to answer the critical questions for logistics and maintenance: "What is the probability of failure before a specific time t?" and "What is the mean time to failure (MTTF)?" This era established the mathematical bedrock

of reliability, providing the tools to quantify the observable randomness of failures, even without a deep understanding of their underlying physics.

2.2. Key Methodologies: Population-Based Statistical Modeling

To quantify the reliability of systems under aleatory uncertainty, the Statistical Era developed a rich toolkit of methodologies. These techniques were not arbitrary; each was based on specific assumptions about the underlying failure process from different types of data. The main task of early reliability engineering was to create a mathematical model for the random variable t, the time to failure. This involved fitting probability distributions to empirical data.

2.2.1. Exponential Distribution: Modeling Random Failures for Electronic Systems

The simplest and arguably most influential model of the Statistical Era is the exponential distribution. Its core principle is the assumption of a constant failure rate λ, which gives the model a unique "memoryless" property, i.e., the probability of a component failing in the future is completely independent of how long it has already been in service, meaning it is not subject to wear-out [34].

The application value of this model was immense, particularly for the burgeoning field of electronics reliability at that time. It provided the first rigorous mathematical description of the "useful life" period of the classic bathtub curve, where failures are caused by random external events like voltage spikes or thermal shocks rather than by intrinsic degradation. Its elegant simplicity made it the default model for complex electronic systems for a crucial reason: the Central Limit Theorem's analogue for reliability suggests that a system comprising many different components, each with its own failure mode and lifetime distribution, will exhibit a system-level failure rate that approximates a constant rate [38].

This principle became the mathematical engine behind early reliability prediction standards, e.g., MIL-HDBK-217F ("Reliability Prediction of Electronic Equipment"). The standard's core methodology was built on the assumption that individual electronic components (resistors, capacitors, integrated circuits, etc.) each followed an exponential distribution with a constant failure rate [39]. The handbook provided extensive tables of base failure rates λ_b for thousands of component types. To calculate the predicted failure rate for a specific component in its operational environment, engineers would use a multiplicative model like

$$\lambda_p = \lambda_b \cdot \pi_T \cdot \pi_E \cdot \pi_Q \cdots \pi_n \tag{3}$$

where λ_p is the predicted failure rate; λ_b is the base failure rate from the handbook; and π_T, π_E, π_Q, $\cdots$, and π_n are different factors (e.g., temperature, environment, quality level, etc.) that adjust the base rate for operational stress.

This standard embodied the failure-centric and decomposition logic of the era [35]. The methodology assumed that a system's overall failure rate λ_{system} could be approximated by summing up that of its individual components $\lambda_{i,p}$. The approach followed a simple additive model:

$$\lambda_{system} = \sum_{i=1}^{n} N_i \cdot \lambda_{i,p} \tag{4}$$

where N_i is the number of the i-th component and $\lambda_{i,p}$ is the predicted failure rate of it after adjusted with various stresses.

Although considered less accurate for complex microelectronics, the fundamental approach of MIL-HDBK-217 continues to be applied, particularly for legacy systems and electromechanical components. Its methodology codified the constant failure rate assumption and the decomposition philosophy that defined the Statistical Era, and its influence persists in modern reliability engineering [35].

2.2.2. Weibull Distribution: A Flexible Model for the Full Lifecycle

The primary limitation of the exponential model was its inability to account for wear-out or infant mortality. The breakthrough in life data analysis was the widespread adoption of the Weibull distribution, a remarkably flexible model that could describe all three phases of the bathtub curve. Its core principle lies in the inclusion of a shape parameter β, which allows the failure rate to change over time. The probability density function (PDF) of the Weibull distribution is given by

$$f(t;\beta,\eta) = \frac{\beta}{\eta}\left(\frac{t}{\eta}\right)^{\beta-1} e^{-(\frac{t}{\eta})^{\beta}} \quad (5)$$

where t is the time to failure, η is the scale parameter and β is the shape parameter.

By estimating β from data, engineers could gain physical insight into the dominant failure mode of a population [40]. The scale parameter η represents the spread range of the component lifetime. What is more, some researchers put forward a three-parameter Weibull distribution by involving a location parameter, which is used to mark the minimum lifetime of a component [41]. With the combination of those parameters' estimation, engineers could make a more precise prediction on component reliability.

The application value of the Weibull distribution was its diagnostic power. For example, the calculation of the "B-Life", such as the B10 life, which is the time at which 10% of the population is expected to have failed [34]. This metric became a standard for specifying design life and comparing the durability of competing component designs from different suppliers [41]. In essence, the Weibull analysis transformed reliability from a simple exercise in counting failures into a predictive and diagnostic science, providing a powerful toolkit for making engineering and business decisions throughout the Statistical Era.

While the Exponential and Weibull distributions were the workhorses of the era, other models were applied for specific scenarios, as summarized in Table 3.

Table 3. Other statistical models for reliability and uncertainty analysis.

Model	Core Principle and Application Value	References
Normal dist.	Primarily models aleatory variability in physical parameters (e.g., manufacturing dimensions, material strength, and electrical resistance). As a lifetime model, it is limited to pure wear-out phenomena where failures cluster very tightly around a mean with low variance.	[42,43]
Lognormal dist.	Model's time to failure for degradation processes resulting from many small, independent, multiplicative effects. Crucial for modeling wear-out in semiconductor devices, bearing fatigue, and some forms of material corrosion. It is often the primary alternative to the Weibull distribution for wear-out analysis.	[44–46]
Binomial dist.	Models the number of failures in a fixed number of n trials. It is the statistical foundation for reliability demonstration testing and is used to determine the sampling size under an acceptable confidence level.	[47,48]
Poisson dist.	Models the number of discrete events occurring over a fixed interval of time, area, or volume. Essential for Statistical Process Control (SPC) in manufacturing to monitor and control the rate of non-conformities, such as defects per square meter of a composite layup.	[49,50]
Gamma dist.	A flexible distribution that can model waiting times for a series of events. It is a generalization of the exponential distribution and is used to model the time to the k-th failure in a repairable system or for systems with standby redundancy.	[51–53]

2.2.3. System Reliability Modeling: From Components to Systems

Once the reliability of individual components was obtained, the central task became predicting the reliability of the whole system. The primary tool developed for this purpose was the Reliability Block Diagram (RBD), a graphical method for representing the logical connections between components and their impact on overall system success [52]. This framework gave rise to a sophisticated toolkit of models for evaluating various system architectures.

The foundational configurations were series and parallel models. The series model represents that system failure will occur if there is any single component failure exists. Conversely, the parallel model means that a system can maintain operation if any individual part is still working. These two simple models were rarely sufficient on their own but served as the essential building blocks for analyzing more complex architecture. For example, recent reliability analyses of eVTOL electric propulsion systems model the components within a single propulsion unit (motor, controller, and propeller) in a series model, while the multiple, independent propulsion units are modeled in a parallel model to represent the system's overall fault tolerance [54].

When reliability criteria are extended to more sophisticated redundancy schemes, the k-out-of-n model was developed to analyze systems with partial redundancy, which function as long as at least k of total n components are operational. The application value of this model is immense for fault-tolerant design, and it remains a cornerstone for modern systems nowadays. For example, the k-out-of-n model is used in advanced avionics based on majority voting for assessing the reliability of battery packs in electric aircraft, where the pack is regarded as functional as long as a minimum number of its many cells are operational [55,56]. Further refinements led to specialized models like the consecutive-k-out-of-n model, which is particularly suited for systems with a linear or circular topology where the failure of several adjacent components is the critical failure mode. The applications could be found in telecommunication relays, sensor arrays or phased array radar [57,58].

Another critical area of reliability modeling involved standby systems, which provided a more nuanced and often more efficient form of redundancy than simple parallel operation. This methodology was crucial because it acknowledged that backup components do not always need to be fully active, leading to significant trade-offs between system availability, power consumption, and lifecycle reliability. The models were categorized as hot/cold/warm standby based on the operational state of the units. If the backup unit is fully powered and running in parallel with the primary unit, then it is a hot standby model. It is used as an instantaneous takeover mechanism [59]. On the contrary, if the backup unit is completely powered off and offline, it is called a cold standby. This model is often used in areas with power-constrain and long-duration missions where long-term reliability is prioritized over instantaneous availability. Between the above two, there is a warm standby model, in which the backup unit is powered on but operates in a low-power or idle state, with only essential functions active. These standby models provided engineers with a sophisticated framework to design redundancy architectures tailored to the specific safety, power, and longevity requirements of a given application, representing a significant step forward in the practical application of reliability theory. For example, a redundant Inertial Reference Unit (IRU) in a commercial airliner is often kept in a warm standby state, because a cold IRU can take several minutes to warm up its gyroscopes and complete its alignment process, which is too long for many in-flight emergency scenarios; and a hot standby IRU would consume significant power and generate excess heat [60].

2.3. Limitations: Unable to Explain Causality of Failure

The statistical paradigm was not an academic curiosity. It was the engine of the 20th century's quality and reliability revolution. Its application value was immense in industries defined by mass production and logistical control, where the historical context of available data and limited computational power made it the ideal toolset. In the aerospace and defense sectors, these methods provided a quantitative basis for maintenance planning, enabling the optimization of spare part inventories and operational availability [61]. In manufacturing, statistical process control and acceptance sampling transformed quality from a subjective art into a quantifiable science, becoming the contractual language between suppliers and manufacturers [62].

Despite its transformative successes, the statistical paradigm was constrained by three fundamental limitations inherent in its "black box" nature. First, it was acausal, offering no insight into the physical root causes of failure and thus providing little prescriptive guidance for design improvement. Second, its heavy reliance on large failure datasets rendered it ineffective for novel designs or high-reliability systems where failure data is, by design, sparse or non-existent [63]. Finally, its system-level models were predicated on an assumption of statistical independence, making them inherently vulnerable to Common Cause Failures (CCFs) that could defeat redundancy and cause systemic collapse [64]. These limitations collectively necessitated a new paradigm that could move beyond empirical observation to a causal, physics-based understanding of why systems fail.

3. The Physics-of-Failure Era: Modeling Causal Chains of Failure

3.1. Core Philosophy: Opening the Black Box for Proactive Design

The limitations of the purely statistical paradigm created a compelling need for a more fundamental approach to reliability. This led to the emergence of the Physics-of-Failure (PoF) paradigm in the 1980s, a movement that represented a profound philosophical shift from reactive observation to proactive, science-based engineering. The central tenet of PoF is that degradation and failure are not merely random events but are deterministic processes governed by the laws of physics and chemistry, which can be understood, modeled, and ultimately, prevented [65]. This era "opened the black box," moving the focus of reliability engineering from empirical evaluation to a causal understanding of failure mechanisms.

The new objective was no longer simply to predict a population's failure rate, but to prevent failure from occurring in the first place through robust design. As pioneered by researchers from the Center for Advanced Life Cycle Engineering (CALCE) at the University of Maryland, the PoF approach advocates for a "know your failure mechanism" methodology [66]. Instead of asking "How long until it fails?", the PoF engineer asks "How does it fail, and what design choices can avoid it?" This transformed reliability from a supporting statistical discipline into an integral part of the core engineering design process, influencing choices in material selection, structural geometry, thermal management, and manufacturing processes [67].

This paradigm also brought a new level of sophistication to the handling of uncertainty. While still primarily concerned with aleatory uncertainty, PoF modeled it at a much more micro level. Instead of treating the time to failure itself as the primary random variable, this approach identified the physical parameters of a degradation model as the sources of randomness. For example, in a solder joint fatigue model, the uncertainty in reliability is a function of the aleatory variability in factors like the ambient temperature cycle ΔT, the device's coefficient of thermal expansion α_{comp} and α_{sub}, and the material fatigue properties ε_f. This relationship is illustrated conceptually in Figure 1.

By propagating these input uncertainties through a physics-based model, engineers could predict the distribution of lifetimes for a new design before a single prototype was

built, a capability that was impossible under the purely statistical paradigm [68]. This proactive, science-driven philosophy allowed reliability to be "designed in" at the early age of development, representing a monumental leap forward in the engineering of robust and durable systems.

It is important to note that the "white-box" philosophy of this era extends beyond the material physics of hardware degradation. In a broader sense, this paradigm represents a shift toward phenomena-based causal modeling. The core objective is to map the specific causal chains that lead to adverse outcomes, whether those chains are rooted in metal fatigue or operational sequences.

Under this broadened definition, techniques such as Probabilistic Risk Assessment (PRA) and Probabilistic Safety Assessment (PSA) represent the mature stage of this era. Unlike pure statistical extrapolation, PRA/PSA attempts to model the logical sequence of events—including human errors and organizational factors—that lead to system failure. By utilizing Event Trees and Fault Trees, these methods introduced a structured way to analyze the "physics" of accident scenarios. However, even these advanced causal models typically rely on the assumption of linear causality and component independence. They model human–machine interaction as a probabilistic failure of a "human component", a simplification that eventually proved insufficient for the complex, non-linear cognitive coupling found in modern complex systems.

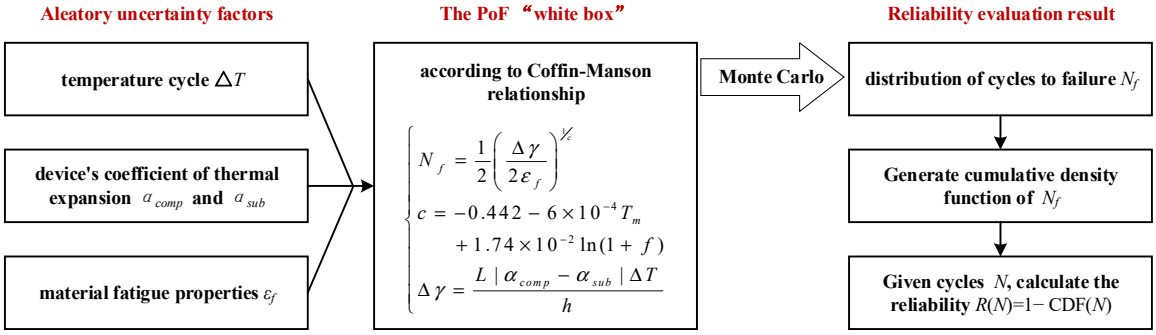

Figure 1. Illustration for using PoF modeled to obtain the reliability of a specific failure mode.

3.2. Key Methodologies: Physical Modeling and Logical Analysis

A science-based understanding of individual failure mechanisms is the necessary foundation of the Physics-of-Failure paradigm. However, knowledge of how a single failure mode will occur is insufficient for assessing the safety and reliability of the whole component, or a system containing various components. Therefore, the PoF methodology rests on two complementary pillars: on one hand, the physical modeling of how components degrade and fail; and second, the logical analysis of how those individual component failure modes propagate and combine to affect overall system performance. On the other hand, to bridge this gap between the micro-level physics of a part and the macro-level functionality of the system, the PoF era relied heavily on structured and decomposed safety and reliability analysis techniques.

3.2.1. Physics-Based Failure Mechanism Modeling

The nature of the PoF approach is the creation of mathematical models that describe the relationship between stress, material properties, and degradation over time. This requires a sophisticated understanding of the dominant failure mechanisms in the target application, and the core principle is to model the physical processes—such as mass transport, charge injection, or defect generation—that lead to component failure [66,69].

For instance, in aerospace electronics, a primary concern is the failure of solder joints due to thermal cycling. Rather than just observing when a joint fails, the PoF approach models the cyclic plastic strain induced by the mismatch in thermal expansion coefficients between a component and the circuit board. This strain is then linked to the number of cycles to failure using a model like the Coffin–Manson equation, allowing engineers to predict lifetime based on material properties and the expected operational temperature swings [70]. Similarly, for mechanical structures like turbine disks or airframe components, the focus is on modeling fatigue crack propagation. Paris's Law provides the physical basis, relating the rate of crack growth per cycle da/dN to the stress intensity factor range ΔK at the crack tip as

$$\frac{da}{dN} = C(\Delta K)^m \tag{6}$$

By integrating this equation, engineers can predict the number of cycles required for a small, initial crack to grow to a critical size, forming the basis of damage-tolerant design and establishing scientifically justified inspection intervals [71]. However, this deterministic model represents an idealized case. In practice, the PoF paradigm explicitly handles aleatory uncertainty by treating the parameters of its physical models not as single-point values, but as random variables described by statistical distributions. For the Paris's Law model, the material constants C and m, the initial crack size a_0, and the stress intensity factor range ΔK are all treated as distributions to account for material variability, manufacturing imperfections, and load fluctuations. By propagating these input uncertainties through the physics-based model, typically using Monte Carlo simulation, the output is not a single lifetime value, but a full probability distribution of the expected cycles-to-failure, from which the reliability function $R(N)$ can be directly derived, as shown in Figure 1.

This principle of combining deterministic physical laws with probabilistic inputs to account for aleatory uncertainty is the main characteristic of the PoF era. The field of physics-of-failure modeling contains many branches, but for aerospace applications, a core set of models has been established to address the most critical failure mechanisms in mechanical structures and microelectronics. As detailed in reference [72], these models provide the foundation for proactive, science-based reliability design. The most critical of these models is summarized in Table 4.

Table 4. Key physics-of-failure models for typical failure models in aerospace reliability analysis.

Failure Mode	Model Purpose and Application	Key Uncertainty Factor	Model formula	References
Mechanical Fatigue	To predict the number of cycles to failure in metallic structures (e.g., airframe, engine disks) under cyclic stress. Essential for damage-tolerant design and setting inspection intervals.	material constants C, m initial crack size a_0 stress intensity factor range ΔK	Paris's Law $\frac{da}{dN} = C(\Delta K)^m$	[73,74]
Electro-migration	To predict the Mean-Time To Failure (MTTF) of metallic interconnects in integrated circuits due to the "electron wind" effect. Critical for avionics processor and ASIC reliability.	current density J temperature T activation energy E_a material constant A current density exponent n	Black's Equation $MTTF = AJ^{-n}e^{(\frac{E_a}{KT})}$	[69,75]
Hot Carrier Injection	To predict transistor lifetime or performance degradation due to high-energy carriers damaging the gate oxide interface. A primary concern for deeply scaled digital logic.	substrate current I_{sub} drain current I_d drain voltage V_{ds} technology-dependent constants	Substrate Current Power Law $\tau \cdot I_d = C\left(\frac{I_{sub}}{I_d}\right)^{-m}$	[76,77]

Table 4. *Cont.*

Failure Mode	Model Purpose and Application	Key Uncertainty Factor	Model formula	References
Negative Bias Temperature Instability	To model the threshold voltage shift in pMOS transistors, which degrades performance over time. A critical reliability issue in modern avionics and processors.	time t temperature T electric field E_{ox} material/process constants.	Reaction-Diffusion Model $\Delta V_{th} = Ae^{(-\frac{E_a}{kT})}t^n$	[78,79]
Time-Dependent Dielectric Breakdown	To predict the time-to-breakdown of the thin gate oxide insulator in a MOSFET. A fundamental lifetime limiter for all modern integrated circuits.	electric field E_{ox} temperature T activation energy E_a field acceleration factor γ	Thermochemical Model $t_{BD} = A_0 e^{-\gamma E_{ox}} e^{(\frac{E_a}{kT})}$	[80,81]

PoF paradigm remains one of the most active and vital methodologies in modern reliability engineering for its unique ability to couple the abstract reliability requirement with practical and designable parameters, which allows reliability to be systematically designed-in from the early stages of development. For instance, in aerospace avionics, PoF models are essential for ensuring the durability of integrated circuits and power electronics under extreme thermal cycling and vibration environments [82]. For developing eVTOLs, PoF is critical for predicting the lifetime and safety of high-density lithium-ion battery packs by modeling degradation mechanisms such as dendrite growth and solid-electrolyte interphase (SEI) layer formation [83]. It is also fundamental to the structural integrity of modern composite airframes, where models for fiber breakage, delamination, and moisture ingress are used to ensure long-term durability [84].

3.2.2. Structured System Safety and Reliability Analysis

With the understanding of individual failure mechanisms, engineers need to translate this micro-level physical knowledge into macro-level system reliability insights with techniques like Failure Modes and Effects Analysis (FMEA) and Fault Tree Analysis (FTA). Since FMEA and FTA need to answer how those individual component failures propagate and combine to affect overall system performance, these methods could be used only when the logical structure between different levels of items had been established.

FMEA is a bottom-up methodology that systematically explores the consequences of failure. The process begins at the component level, asking the fundamental question: "What happens if this component fails in this specific way?" [85]. For each component in a system, engineers list all credible failure modes (e.g., a resistor fails open; a hydraulic valve fails closed). They then trace the effects of each failure mode upwards through the system's architecture. It means that the engineer needs to establish the relationship among the local effect (e.g., loss of signal), the subsystem effect (e.g., control channel goes offline), and the end effect on the overall system (e.g., loss of flight control). By assessing the severity, probability of occurrence, and detectability of each failure mode, a risk priority number (RPN) can be calculated to prioritize mitigations. FMEA's primary value is as a proactive design tool, forcing a rigorous and systematic consideration of potential failures early in the development process [86].

FTA, in contrast, is a top-down methodology that begins with a known system-level hazard and works backward to identify its root causes. The analysis starts with a single, undesired "top event" (e.g., "Unexpected Engine Shutdown") and asks the question: "What component failures or events, alone or in combination, could lead to this hazard?" [87]. The analyst decomposes the top event into a series of intermediate events linked by logical gates (primarily AND and OR gates) until reaching the "basic events", i.e., the fundamental root causes, which are typically individual component failures. The great power of FTA lies in

its ability to identify complicated combinations of failures and to be easily quantified. If the probabilities of the basic events are known, the probability of the top-level hazard can be calculated. The analysis also yields "minimal cut sets," which are the smallest combinations of basic events that will guarantee the top event occurs, thereby highlighting the system's most critical vulnerabilities [88].

Table 5 lists some main characteristics of FMEA and FTA for their application. These methods excel when the system architecture allows for clear, hierarchical, and well-defined structural decomposition. Therefore, the efficacy of these methods is heavily dependent on the analyst's engineering experience and their ability to foresee all credible failure modes and interaction pathways. In essence, FMEA and FTA are powerful tools for analyzing systems where the primary uncertainty is the aleatory timing of known failure modes. They are fundamentally ill-equipped to handle the epistemic uncertainty associated with unknown or emergent failure modes that arise from complex, tightly coupled interactions, a defining characteristic of modern, software-intensive aerospace systems [10].

Table 5. The main characteristics of FMEA and FTA.

Attribute	FMEA	FTA
Logic	Bottom-Up: Forward-chaining from cause to effect.	Top-Down: Backward-chaining from effect to cause.
Guiding question	What happens if this component fails?	How can this system hazard happen?
Purpose	To explore the effects of potential component failures and identify their severity for risk prioritization.	To identify all credible combinations of failures (minimal cut sets) that lead to a specific top-level hazard.
Key output	A structured table listing failure modes, their effects, severity, and risk priority number (RPN).	A logical tree diagram, a list of minimal cut sets, and a calculated probability for the top-level event.
Core Assumption	System hazards are the result of the summed or sequential effects of individual component failures.	System hazards can be represented as a Boolean combination of basic component-level failure events.

3.3. Limitations: When the Whole System Is Beyond the Sum of Its Parts

The very strength of the PoF paradigm—its intense, disciplined focus on component physics—is simultaneously its greatest weakness in the face of modern system complexity. While PoF excels at modeling components, it struggles with systems that are beyond the sum of their parts. In the context of this review, this limitation refers to the inability of reductionist methods to identify emergent properties. For highly integrated avionics or autonomous platforms, hazardous behaviors often arise from the correct operation of components interacting in unforeseen ways (e.g., a software logic trap), rather than from physical breakage. Since the system's risk profile is dominated by these interactional properties, simply aggregating the reliability of individual parts fails to capture the true safety state of the whole system. For an advanced flight control system comprising massive software code, thousands of electronic parts, and complex integrated circuits, the exhaustive, physics-based modeling of every potential failure mechanism for every component is computationally and economically infeasible [89]. This challenge is further compounded in the context of novel materials and advanced packaging, where validated physical models may not even exist [90]. Consequently, PoF is often relegated to analyzing a handful of critical components rather than the system as a whole.

In addition, the techniques are useless to hazards arising from the interactions between correctly functioning components. This conceptual weakness is the critical vulnerability in modern systems. Systems-Theoretic analyses, for example, reveal how fully functional subsystems in unmanned aircraft can interact to create catastrophic risks that FMEA and

FTA cannot identify and analyze [91]. Likewise, increasing cockpit autonomy introduces hazards rooted not in system failure, but in flawed human–system interaction, such as automation-induced mode confusion that can lead to flawed pilot decision-making [92]. In conclusion, the PoF paradigm perfected the analysis of component-based, hardware-centric failures. This success was instrumental in achieving the component reliability necessary for complex systems to exist. However, this triumph inadvertently created systems so intricate that their most significant risks now lie not in the physics of the parts, but in the logic of their interactions. The PoF paradigm, with its decomposed foundation, has reached its conceptual limit.

4. The Prognostics Era: Predicting Failures Through Real-Time Monitoring

While the PoF paradigm provided a robust framework for design-for-reliability, scholars have widely recognized its inability to account for the unique operational histories and environmental stresses that govern the health of individual assets. This fundamental gap between static design models and dynamic in-service reality catalyzed the evolution toward the Prognostics Era. As influential reviews by academics articulate, this represents a philosophical shift from a passive, failure-focused reliability approach to a proactive one centered on real-time performance and health management [93]. Enabled by the proliferation of advanced sensor technologies and data analytics, the new paradigm seeks to understand and predict failures not by applying generalized population models, but by continuously monitoring the specific, evolving health of each system [94]. At its heart, the Prognostics Era, therefore, embodies a fundamental change in the management of uncertainty—from passively quantifying it before deployment to actively reducing it through the continuous assimilation of operational evidence, a transition that has redefined the frontiers of safety and reliability engineering [95].

4.1. Core Philosophy: From Static Uncertainty to Dynamic Health Management

The core philosophy of the Prognostics Era is to reframe reliability engineering from a static and design-related property of a population into a dynamic and manageable attribute of an individual system. This evolution was not merely an incremental improvement but a necessary response to the fundamental limitations in how the PoF paradigm handles uncertainty. While PoF excels at quantifying the aleatory uncertainty inherent in material properties, manufacturing tolerances, and anticipated loads at the design stage, its output is a single, static reliability curve intended for an entire population [96]. For instance, the stress that a component of a system may deviate from profiles assumed in its design phase. This gap between the generalized uncertainty of a population and the specific uncertainty of an individual item is the critical problem that the prognostic philosophy was conceived to solve [97].

Instead of treating a component's lifespan as a fixed probability, prognostics treats it as an evolving state of knowledge that must be continuously updated. The central goal is to leverage real-time sensor data as evidence to progressively reduce the uncertainty by presenting a component's current state and its future degradation trajectory [93]. This new philosophy is enabled by the development of advanced sensing technologies, which provide the high-fidelity data streams necessary for real-time fault detection and degradation monitoring [98]. To be noted, this emergence of the new paradigm is led by the reorientation of the reliability engineering objective. The goal is no longer simply to quantify a predetermined uncertainty distribution at the beginning of life, but to actively control and minimize the uncertainty of physical failure risk throughout the operational life of each specific item. The output of a prognostic system, a quantified remaining usage life (RUL) with its associated probability, is not merely an informational metric, but an actionable input

for dynamic risk management [99]. By providing a high-confidence forecast of impending failure, it enables a direct control action to mitigate the risk before it can be realized [100].

4.2. Key Methodologies: Apply RUL to Predict Failure Trend

The quantification of RUL is the central task of fault prognostics, providing the predictive insights necessary to manage failure trends. There are various methods to achieve RUL based on the employed models. According to a comprehensive survey, these techniques can be classified into three primary families, i.e., physics-based approaches, data-driven approaches and hybrid approaches.

4.2.1. Physics-Based (or Model-Based) Approaches

The physics-based approach to prognostics is indeed a direct application of the PoF paradigm, extending its principles from the design phase into the operational lifecycle of an item. Therefore, the core of this method is to create an explicit mathematical model of a failure mechanism and use real-time sensor data to drive the model's parameters, track its state evolution, and project its state into the future. The key difference is the source of input data. PoF relies on assumed mission profiles, whereas physics-based prognostics relies on actual, measured operational data. For example, in the classic Paris's Law for fatigue crack growth in Equation (6), the stress intensity factor range ΔK is no longer a design assumption but is calculated from real-time strain or vibration sensor data. By integrating this equation using the actual load, an engineer can track the crack's current length a and predict its future growth far more accurately than with static models.

Similarly, by employing the structured system reliability model, the performance of the whole system can be predicted dynamically. Figure 2 shows a RUL dynamic evaluation during system operation. At the beginning, we can calculate the RUL based on the assumed PoF model parameters and arrange a repair at time t_0 as the system's performance is going to hit the threshold. As the system continues to operate and the component i fails at a certain time, the structured system reliability model changes. So, the performance evaluation of the system drops, and we need to adjust the repair time to be t_i. Similarly, if the component j fails at another time, the structured system reliability model changes, and causes the performance evaluation of the system to drop again. Thus, we need to arrange the repair at t_j. This simple example tells us that we can predict the system performance with the structured system reliability model.

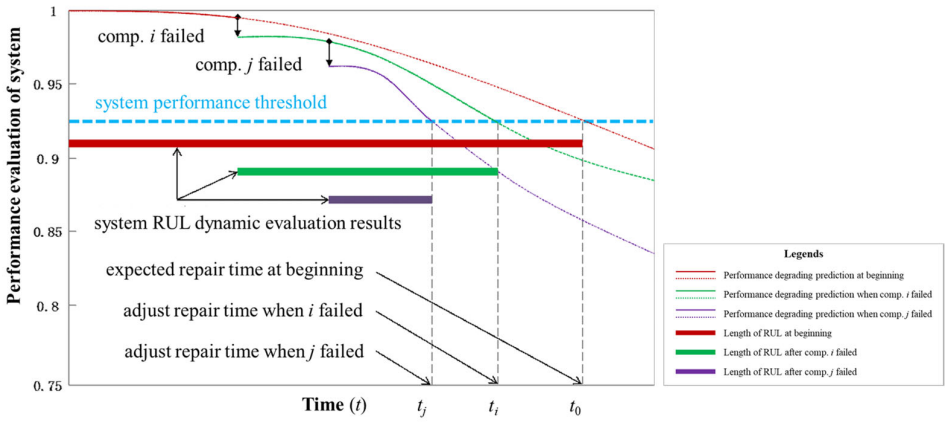

Figure 2. System performance evaluation based on reliability models.

4.2.2. Data-Driven Approaches

With the advancement in Artificial Intelligence (AI) and the growth in data processing capabilities, data-driven approaches have emerged as the most prominent uncertainty quan-

tification in the prognostic paradigm of reliability engineering in recent years. Therefore, a significant portion of contemporary PHM literature is dedicated to developing, refining, and applying these data-driven methodologies. Data-driven approaches for prognostics operate on a principle of causality, which aims to learn and recognize the patterns of what a system's behavior looks like before it fails by extracting degradation laws from a large amount of historical data. Thus, the primary objective is to train a complex, non-linear mapping function $f(\mathbf{X})$ to establish the relationship between a time-series sensing data $\mathbf{X} = \{X_t, X_{t-1}, \ldots, X_1\}$ and the prediction of RUL. The literature highlights a variety of AI techniques for this task. In this paper, we will review some of the most popular methods.

Firstly, we will illustrate how to apply the recurrent neural network (RNN) and its variants for RUL prediction. RNN is the main method of deep learning for time-series prognostics due to their inherent ability to model sequential data. An RNN processes a sequence step-by-step, maintaining an internal hidden state h_t that acts as a memory, capturing information from all previous steps. This is achieved through a recurrent connection:

$$h_t = \sigma(W_h h_{t-1} + W_x x_t + b) \tag{7}$$

where h_{t-1} is the hidden state from the previous time step, x_t is the current input, W and b are learned weights and biases, and σ is an activation function. In this equation, h_t represents the expected system state considering all the historical parameter values. Therefore, we can train an output layer:

$$\text{RUL} = W_y h_t + b_y \tag{8}$$

where W_y and b_y are the pretrained weights and biases of the output layer neural network. The RNN only works well for the simple system because it suffers from the vanishing gradient problem, making it ineffective at learning long-term dependencies. To overcome this, advanced variants like the Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) were developed [101]. These architectures introduce sophisticated "gating mechanisms" (e.g., forget, input, and output gates in LSTM) that allow the network to selectively remember relevant information and forget irrelevant data over long periods. This makes them exceptionally powerful for RUL prediction, where early-life sensor readings can be critical for late-life failure forecasting [102]. In practice, when researchers refer to using RNN for sequence modeling, like RUL prediction, they almost always mean using LSTM or GRU.

Secondly, we will illustrate how to apply the convolutional neural network (CNN) for RUL prediction. Although CNN was originally designed for image processing, reliability researchers employ it for PHM by treating time-series parameters as one-dimensional data. The basic steps for applying CNN to RUL prediction begin with transforming the sensing data into segments $\mathbf{X} = [x_1, x_2, \ldots, x_L]$ with a fixed length L. Then, we need to denote several filters (or kernels) $\mathbf{W} = [W_1, W_2, \ldots, W_k]$, used to capture the features in the time-series data, e.g., a high-frequency spike or a specific oscillation. For each position t in an arbitrary segment, a filter needs to measure how well the signal at position t matches the filter's feature with the following equation:

$$c_t = f\left(\sum_{i=1}^{k} (W_i \cdot x_{t+i-1}) + b\right) \tag{9}$$

where c_t is the feature map of a single filter at position t, W_i is the i-th weight of the filter, x_{t+i-1} is the data that the i-th filter is currently overlapping, b is a learnable bias term, and f is an activation function which we usually employ $f(z) = \max\{0, z\}$. After pooling and

flattening operations, the multi-dimensional tensor is transferred into a one-dimensional vector $V_{flattened}$, which preserves all the learned features in a format that is suitable for the final prediction stage. Thereafter, with the pretrained parameters W_{fc} and b_{fc}, the RUL can be predicted by the following equation:

$$\text{RUL} = W_{fc} \cdot V_{flatten} + b_{fc} \tag{10}$$

where W_{fc} is the weight of the features in $V_{flattened}$ on influencing the RUL, and b_{fc} stands for the base RUL prediction. We can regard the dot product term $W_{fc} \cdot V_{flatten}$ as the accumulated effect of all the features on RUL, i.e., it is an adjustment of the base RUL. Therefore, the main advantage of employing CNN for RUL prediction is the ability to perform automatic hierarchical feature extraction [103,104], regardless of the reliance on manual identification of degradation features.

Thirdly, we will illustrate how to apply the transformer-based models for RUL prediction. The transformer was originally developed for natural language processing. It becomes a new frontier in RUL prediction due to its unique self-attention mechanism, which allows the model to weigh the importance of all time steps in a sequence simultaneously when making a prediction for a given point [105,106]. Let $\mathbf{X} = [x_1, x_2, \ldots, x_L]$ be the sequence of data, after the positional encoding operation, which contains both sensing and position information. Then, for each time step's embedding in $\mathbf{X}$, its importance (i.e., attention) to others can be obtained by the following equation:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{11}$$

where QK^T creates a similarity matrix where each entry represents how "relevant" one time step is to another, d_k is the dimension of vectors K, the softmax function is applied to convert the scores into a set of positive weights that sum to 1 and V is the value matrix. Equation (11) produces a new sequence of weights for each time step over the others. After the sequence has passed the multi-head attention, through one or more transformer blocks, we have a final sequence of output embeddings $V_{features}$ that contains a comprehensive summary and final judgment of the current health status of the equipment. Similarly, with the pretrained parameters W_{fc} and b_{fc}, we can get the RUL prediction by

$$\text{RUL} = W_{reg} \cdot V_{features} + b_{reg} \tag{12}$$

The transformer model for RUL prediction performs well at capturing complex, long-range dependencies, making it a state-of-the-art method, particularly for long and intricate time-series data [106].

Lastly, to leverage the complementary strengths of different models, hybrid architectures have become a dominant trend in state-of-the-art RUL prediction [107,108]. The most common and effective hybrid model is the CNN-LSTM architecture. In this structure, the CNN acts as a powerful feature extractor, processing raw and high-frequency sensor data segments to produce a condensed, informative feature representation. This sequence of learned features is then fed into an LSTM, which models the temporal evolution of these features to make the final RUL prediction [109]. This approach synergistically combines the spatial feature learning of CNN with the temporal sequence modeling of LSTM, often yielding superior performance compared to either model used in isolation [110].

4.2.3. Physics and Data Integrated Hybrid Approaches

Hybrid approaches have emerged as the most promising direction in modern prognostics, seeking to synergistically combine all available sources of knowledge to overcome

the respective limitations of pure physics-based or data-driven models [111]. This fusion of first-principles knowledge with the powerful function-approximation capabilities of machine learning is proving to be a critical strategy for developing trustworthy RUL prediction frameworks for safety-critical systems [112]. Several distinct integration strategies have been prominently featured in the recent literature. In this paper, to the best of our knowledge of the research frontier, we will only review some of the most popular physical and data hybrid approaches.

Firstly, we will review the state-space filtering model for RUL evaluation. This model is primarily implemented using sequential Monte Carlo methods like the Kalman Filter (KF) and Particle Filter (PF), which separates the roles of physics and data into a recurring two-step cycle, i.e., prediction and correction [113,114]. In the prediction step, we employ a physics-based state transition model to forecast the evolution of the system's hidden health state from one time step to the next. This is governed by the state equation:

$$x_k = f(x_{k-1}, u_k) + w_k \tag{13}$$

where x_k represents the state vector at each time epoch (e.g., crack size or battery impedance), $f(\bullet)$ is the physics-based state transition function, u_k represents any control inputs, and w_k is the process noise accounting for model uncertainty [115]. In the correction step, we employ a measurement function $h(x_k)$ to incorporate real-time sensor data, z_k, which is linked to the state by the following equation:

$$\hat{z}_k = h(x_k) + v_k \tag{14}$$

where v_k is the measurement noise [116]. The filter compares the actual measurement z_k with the predicted measurement $\hat{z}_k$ and uses the discrepancy to update the state estimate. This predict–correct cycle allows the physics model to provide a robust and interpretable structure, while real-time data continuously corrects for model inaccuracies and tracks system-specific degradation [117].

Secondly, we will introduce a rapidly emerging technique called physics-informed neural network (PINN), which embeds physical laws directly into the loss function for a neural network pretraining [118]. By doing so, it forces the data-driven model's predictions to be physically consistent, even in regions with sparse data. The core principle is the modification of the standard neural network loss function. Traditionally, the loss function L_{data} is simply the mean squared error (MSE) between the network's predictions and the training data. However, PINN adds a second term $L_{physics}$ that penalizes the network if its output violates a known governing physical law. The combined loss function is denoted as

$$L_{total} = L_{data} + \lambda L_{physics} \tag{15}$$

where $L_{physics} = \sum_{i=1}^{N} [r(t_i, x_i)]^2 / N$ is the mean squared residual of the governed by the partial differential equation (PDE) according to specific physical law, $r(t_i, x_i)$ is the physics residual given data x_i at time step t_i, and λ is a hyperparameter that balances the contribution of the data-driven and physics-based loss terms. By minimizing this composite loss, the network learns a solution that both fits the observed data and adheres to the fundamental principles of physics [119]. This is particularly valuable for prognostics in the area where failure data is rare, as the physics-based loss regularizes the solution and prevents unrealistic predictions [118].

Lastly, we will introduce the physics-informed data augmentation approach in RUL prediction. The basic principle of this method is to synthetically generate data that is physically consistent with degradation trajectories for neural network training. This methodology

is typically a two-stage process [120]. The first stage involves creating a robust digital model of the system in its healthy state. This is often achieved by applying system identification techniques to the limited amount of real, healthy operational data available, resulting in a validated dynamic model of the system's nominal behavior. This digital model can then be used to generate an augmented and enriched dataset of nominal operations under various conditions. The second stage involves injecting a physics-based degradation model into a specific component of this validated digital system model. This degradation model is a mathematical representation of a known failure mechanism. For instance, to simulate the degradation of an actuator valve due to increased friction, a well-established stiction model needs to be injected. One such model is described by the following equation:

$$x_k = \begin{cases} x_{k-1} + (e_k - sign(e_k) f_D), & if\ |e_k| > f_s \\ x_{k-1}, & if\ |e_k| \leq f_s \end{cases} \quad (16)$$

where x_k is the valve position, e_k is the error between the command and position, and f_s and f_D are the static and dynamic friction parameters. By systematically increasing a parameter like f_s over simulated time, a gradual and physically realistic degradation process is induced, from healthy operation ($f_s = 0$) to complete failure. Other classic degradation models, such as Paris's Law for fatigue or the Arrhenius model for chemical degradation, can be similarly injected depending on the component being studied. This approach not only alleviates the data scarcity problem but also enhances the transparency and trustworthiness of the subsequent AI-based predictions.

4.3. Limitations: Distribution-Shift and OOD

While the Prognostics Era introduced the critical dimension of time, its fundamental reliance on data-driven inductive inference creates a new class of safety risks. The validity of any RUL prediction rests on the stationarity assumption, i.e., the future operational data must follow the same statistical distribution as the training data. For modern complex systems, this assumption is frequently violated in three ways. First, deep learning models often exhibit "overconfident but wrong" behavior in Out-of-Distribution (OOD) scenarios. For instance, an eVTOL encountering novel vibration patterns may be confidently misclassified as a safe state, leading to a dangerous failure. Second, the distribution-shift occurs as underlying physical relationships evolve due to aging or environmental seasonality, rendering models trained in specific historical conditions inaccurate over time. Third, even physics-informed data augmentation suffers from a simulation-to-reality gap, where models overfit to simplified physics and fail against real-world stochasticity. These limitations reveal a fundamental truth for the PHM models that it is impossible to maximize a training dataset to cover the infinite state space of an open environment. Consequently, relying solely on reducing epistemic uncertainty through better prediction is a strategy of diminishing returns. The engineering paradigm must therefore pivot from asking "how to predict better" to "how to remain safe when prediction inevitably fails", providing the logical imperative for the UC framework discussed next.

5. The Resilience Era: Focusing on Mission Success Under Uncertainty

5.1. Core Philosophy: Operating Beyond the Limits of Knowledge

The transition to the Resilience Era is not merely a change in technique but a fundamental epistemological shift. It begins with the engineering community's common admission that for modern, software-defined, and autonomous systems, the "complete knowledge" of the system's behavior is no longer attainable. This philosophical realization is codified in the evolution of critical aerospace standards. A prime example is the significant terminol-

ogy update in SAE ARP4754B [29]. The standard deliberately replaces the previous concept of "unintended function" with "unintended behavior." This change represents a profound acknowledgment of complexity, i.e., while a "function" implies a discrete, identifiable design element that can be simply verified as correct or incorrect, "behavior" encompasses the emergent, dynamic, and continuous outcomes of system interactions. By adopting "behavior," the industry formally admits that hazardous states may arise not just from discrete design errors, but from complex systemic interactions that were never explicitly "functionalized" in the requirements. This signifies that we can no longer simply "debug" a system into safety; we must instead manage its emergent behavior.

5.1.1. The Limit of Predictability and the "State-Space Explosion"

In the traditional reliability paradigms (Statistical and PoF), the underlying assumption was that the system is deterministic and that all critical failure modes could be enumerated, tested, and mitigated. However, this assumption collapses under the weight of the "state-space explosion" inherent in modern avionics and Urban Air Mobility (UAM) architectures.

For a legacy electromechanical system, the number of failure states was finite and manageable. In contrast, modern autonomous systems driven by AI/ML algorithms possess a virtually infinite state space. Koopman and Wagner [2] demonstrated that to statistically validate the safety of an autonomous vehicle to a level comparable to human pilots (10^{-8} failures per hour), using test-driving alone would require billions of miles of testing and taking tens or hundreds of years. This creates a "Validation Gap" where empirical testing can only cover a negligible fraction of the operational envelope.

Furthermore, the uncertainty has shifted from "known unknowns" (e.g., component fatigue life) to "unknown unknowns" (e.g., emergent behavior in rare scenarios). As noted in a NASA study on UAM airspace safety, the integration of non-deterministic agents creates a complex adaptive system where hazardous behaviors emerge from the interaction of correctly functioning components [121]. In such systems, the probability of encountering an unforeseen state $x_{unknown}$ is non-zero. Therefore, basing safety solely on the prediction of known failures is mathematically insufficient.

5.1.2. From "Fail-Safe" (Safety-I) to "Safe-to-Fail" (Safety-II)

To survive in this high-uncertainty environment, the engineering objective must migrate from the passive "Fail-Safe" logic of Safety-I to the active "Safe-to-Fail" logic of Safety-II.

- Safety-I (The Absence of Negatives): This traditional view defines safety as a condition where the number of adverse outcomes (accidents/incidents) is as low as possible. It focuses on "bimodal" outcomes: the system either works perfectly or fails.
- Safety-II (The Presence of Positives): As articulated by Hollnagel in his recent works [122], Safety-II defines safety as the system's ability to succeed under varying conditions. It acknowledges that performance variability is inevitable and necessary for adaptation.

In the context of eVTOLs, this shift is critical. A Safety-I approach might design an autopilot to disengage upon detecting a sensor anomaly, handing control back to a pilot. However, in a simplified single-pilot or autonomous UAM operation, sudden disengagement could be catastrophic due to the pilot's loss of situational awareness [123]. A Safety-II approach (Resilience) would instead design the system to maintain functional endurance—perhaps by degrading to a "safe hover" mode using synthetic sensor estimates—thereby ensuring mission success or a safe recovery despite the anomaly.

5.1.3. Regarding Safety as a Control Problem

If we cannot predict every failure, then we must instead constrain the system's behavior. This philosophy relies heavily on the Systems-Theoretic Accident Model and Processes (STAMP) theory developed by Leveson, which has gained renewed urgency in the 2020s for certifying autonomous systems [124]. The central argument is that safety is an emergent property of the system level, not the component level. In complex software-intensive systems, accidents often occur without any component "failure" in the reliability sense. For example, the loss of a flight control system might occur because two software modules, both working exactly as specified in their requirements, interact in a way that drains the batteries [88].

Therefore, the engineer's role shifts from increasing the reliability of individual parts to designing a manageable structure that controls the safety constraints. The system is modeled not as a chain of failure events, but as a dynamic control loop where a controller issues actions to a process based on a process model. Safety is breached when the controller's process model deviates from reality, e.g., the software thinks the aircraft is climbing when it is stalling.

Formally, we define a Safe Envelope Ω_{safe}. The objective of uncertainty control (UC) is to design a control law $u(t)$ such that the system state $x(t)$ remains within Ω_{safe} despite the presence of epistemic uncertainty $\Delta(x)$ and external disturbance $d(t)$

$$\forall t, \{x(t+1) = f(x(t), u(t)) + \Delta(x) + d(t)\} \in \Omega_{safe} \tag{17}$$

This formulation effectively decouples safety assurance from the need for perfect knowledge of $\Delta(x)$. If the control architecture can enforce the boundary of Ω_{safe}, the deep epistemic uncertainty of components becomes manageable. This principle forms the theoretical foundation for the methodologies we will discuss next: STPA and runtime assurance.

5.2. The Strategic Shift: From Uncertainty Quantification (UQ) to Uncertainty Control (UC)

While the previous eras (Statistical, PoF, Prognostics) were obsessed with quantifying uncertainty—calculating the probability of failure more precisely—the Resilience Era recognizes that for complex adaptive systems, quantification alone is a passive exercise that does not guarantee safety, especially for the system with aleatory uncertainty. Thus, the strategic paradigm must shift from uncertainty quantification (UQ) to uncertainty control (UC).

5.2.1. Defining Uncertainty Control: The Safety Envelope

UC is defined not as the elimination of aleatory or epistemic uncertainty—which is often impossible in open environments—but as the active containment of system behavior within a valid safety envelope. In this context, the system is allowed to exhibit complex, non-deterministic, or even "messy" behavior internally, provided that its external physical manifestation never violates the safety envelope.

Mathematically, this is often formalized using Control Barrier Function (CBF), a method that has gained significant traction in safety-critical robotics and aerospace recently [125]. Let the safety envelope E be defined by a super-level set of a continuously differentiable function $h(x)$:

$$\mathrm{E} = \{x \in \mathbb{R}^n : h(x) \geq 0\} \tag{18}$$

where x is the system state (e.g., aircraft position or velocity). UC acts as a filter on the control input u. Even if the primary controller (e.g., an AI agent) requests an unsafe action

u_{AI} due to internal uncertainty, the UC mechanism solves a quadratic program in real-time to find the closest safe control action u^* that satisfies the barrier condition:

$$\begin{aligned} u^* &= \underset{u}{\operatorname{argmin}} \, ||u - u_{AI}|| ^2 \\ s.t. &\frac{\partial h}{\partial x} f(x,u) + \alpha(h(x)) \geq 0 \end{aligned} \tag{19}$$

This equation mathematically guarantees that the system state x will never leave the safety envelope E, regardless of the uncertainty inherent in the AI controller u_{AI} [126].

It should be noted that while u^* represents theoretically optimal safe control action, its analytical existence and real-time solvability in high-dimensional non-linear systems remain significant challenges. In practice, the RTA framework addresses these issues by decoupling the safety monitor from complex online optimization. Instead of attempting to solve for u^* in real-time—which could introduce hazardous computational delays—modern RTA architectures typically switch to a pre-defined, simplified backup controller (e.g., a robust PID or a conservative glide-mode law) once the system approaches the boundary of the safe invariant set. This strategy ensures that even in degraded modes or extreme scenarios where the nominal model may fail, a verified and computationally efficient control action is available to maintain fundamental stability, thereby sidestepping the risks of solver non-convergence or latencies.

5.2.2. Decoupling Assurance from Complexity

The most revolutionary implication of the UC paradigm is the capability to decouple safety assurance from functional complexity. In the traditional certification approach (e.g., DO-178C), safety is assured by verifying the correctness of the entire control logic. However, for a deep neural network (DNN) with millions of parameters, tracing logic is impossible. UC solves this by encapsulating the complex, non-deterministic component within a deterministic safety monitor. This concept is central to ASTM F3269-21 which establishes the architectural standard for run-time assurance (RTA) [127].

- The Complex Core: An AI-based flight controller that optimizes fuel efficiency and passenger comfort. Its internal uncertainty is high.
- The Assurance Layer: A deterministic, physics-based RTA safety monitor that only enforces basic flight envelope limits, e.g., angle of attack $\alpha < \alpha_{stall}$ or G-load < 2.5 g.

While the assurance layer ideally relies on a high-fidelity physics-based model, its practical implementation must account for extreme epistemic uncertainty or heavily degraded modes (e.g., structural damage or total sensor failure). In such marginal cases, where the nominal flight envelope becomes impossible to solve, the RTA architecture is designed to prevent the worst situation from happening. RTA will utilize the fail-safe logic, such as a simplified glide-mode or a neutral stability command, that does not require full system state identification. By prioritizing deterministic survivability over complex performance, the assurance layer remains functional even when the precise system physics are partially unknown.

By validating only the assurance layer (which is simple and physics-based) and not the AI Core, engineers can certify the safety of the aircraft without needing to fully understand the "black box" of the AI. EASA's Artificial Intelligence Roadmap 2.0 explicitly validates this strategy, categorizing it as "W-shaped" development [128]. It allows for the deployment of non-deterministic algorithms by ensuring that their unintended behaviors are intercepted before they propagate to the actuators.

This decoupling is the key that unlocks the future of autonomous systems, allowing innovation in performance (via AI) while maintaining strict guarantees on safety (via UC). Table 6 represents a difference in the UC paradigm from the previous era's UQ

paradigm. For instance, the UQ paradigm regards the system as deterministic and focuses on how to reduce the probability that the unintended function could occur, whereas the UC paradigm admits that the system may be non-deterministic, so it pays attention to how to manage unintended behavior from affecting other parts. Another difference between the two paradigms is their attitude about handling uncertainty. The UQ paradigm mainly considers the aleatory uncertainty, which can be reduced with accumulated testing data. However, the UC paradigm mainly involves epistemic uncertainty, for which we need to design multiple protection mechanisms to stop its occurrence. In addition, during the UQ paradigm era, the key metric that we care about is the failure rate, so we need to verify that the entire system satisfies the total failure rate requirement. But, in the UC paradigm era, we need to measure the resilience capability metric, e.g., the leading time to predict the system performance decline or the recovery time for the system performance bounce back to a certain level. Therefore, we need to verify that the system contains the safety management functions.

Table 6. Comparison of assurance strategies.

Feature	Traditional (UQ Paradigm)	Resilience (UC Paradigm)
Focus	Unintended function	Unintended behavior
Assumption	System is deterministic	System may be non-deterministic
Handling Uncertainty	Reduce it through testing	Contain it through architecture
Key Metric	Failure Rate	Resilience capability
Verification Target	The entire complex system	The safety management functions

5.3. Key Methodologies: STPA and RTA

5.3.1. Designing for Control with STPA

If uncertainty control (UC) is the strategic objective, then Systems-Theoretic Process Analysis (STPA) is the architectural tool designed to achieve it. While traditional methods like FMEA focus on component reliability, STPA focuses on system control, i.e., preventing the system from doing the wrong thing, even when nothing breaks.

1. The Standardized Workflow of STPA

STPA follows a four-step iterative process designed to uncover unsafe interactions [124], as illustrated in Figure 3.

- Step 1: Define Purpose of Analysis.

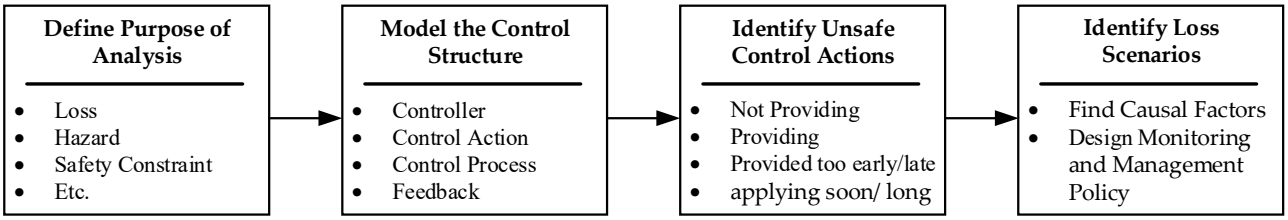

Figure 3. The four-step workflow of STPA.

The process begins by identifying the system-level losses (e.g., loss of life, hull loss) and hazards (system states that lead to a loss). In addition, the high-level safety constraints are derived too.

- Step 2: Model the Control Structure.

Engineers should establish functional feedback loops by defining controllers, actuators, controlled processes, and Sensors. This structure captures the information flow and hierarchy of the system.

- Step 3: Identify Unsafe Control Actions (UCAs).

This is the key technical process of STPA. A UCA is a control action that violates a safety constraint in a specific context. The UCA could originate from four situations. First, the control action is "not provided", so that a hazard may occur. Second, the control action is "provided", so that a hazard may occur. Third, the control action is "provided too early, too late, or out of sequence." Fourth, the control action is "stopping too soon or applying too long".

- Step 4: Identify Loss Scenarios.

Once a UCA is identified, the engineer works backward to find the causal factors, ranging from component failures to inadequate feedback, process model mismatches, or conflicting requirements, that could lead the controller to issue the UCA. Thereafter, with this information, engineers can design some monitoring and management policies to prevent the occurrence of a certain UCA.

2. Handling Interactional Risks in the Design Phase

Since the dominant risks in modern aerospace systems arise from unsafe interactions rather than component failures, STPA, grounded in Leveson's STAMP theory, addresses this by modeling the system not as a chain of events, but as a hierarchical control structure [129]. In the STPA framework, a hazard is not caused by a "failure" but by an unsafe control action (UCA), which occurs when a controller (human, software, or mechanical) issues a command that violates the safety envelope of the controlled process. STPA systematically scans for four types of UCAs:

- A control action required for safety is not provided.
- An unsafe control action is provided.
- A control action is provided too early or too late.
- A control action is stopped too soon or applied too long.

3. Evidence of Superiority: Beyond Component Failure

Recent studies have quantified the advantage of STPA over traditional methods in software-intensive systems. A comparative study applying both FMEA and STPA to an automated aircraft braking system shows that while both methods identified identical hardware failure modes, STPA identified 45% more causal factors, all of which were related to software requirements errors and complex interaction scenarios that FMEA completely missed [130].

- A Brief Case Study: eVTOL Transition Phase

Consider the critical "transition" phase of an eVTOL aircraft (switching from vertical hover to wing-borne flight). The FMEA approach would analyze failures like "Actuator stuck" or "Sensor dead", whereas the STPA approach could identify a UCA, such as the flight control computer (FCC) commands "Push Nose Down" while the altitude is lower than 50 ft. This could be due to a process model mismatch, since the FCC believes the aircraft is higher than it is due to a barometric pressure drift, or it prioritizes air speed over altitude due to a flawed requirement specification.

This ability to catch requirements flaws and mode confusion is critical. In the context of a human–autonomy coupled system, STPA is uniquely capable of identifying hazards where the pilot and the automation have conflicting perceptions of the system state. A research of UAM operations highlighted that "automation surprise"—a classic epistemic uncertainty problem—was the leading cause of UCAs in emergency scenarios, a risk invisible to component-level analysis [131].

4. The Output: From Probabilities to Safety Constraints

Unlike FMEA, which outputs a risk priority number (RPN) or a probability of failure, the output of STPA is a set of rigorous safety constraints. This aligns perfectly with the philosophy that regards safety as a control problem. If a UCA is defined as a tuple of context and action that leads to a hazard

$$UCA = \{(u, x) | \text{action } u \text{ in state } x \Rightarrow hazard\} \tag{20}$$

Then the goal of the STPA process is to generate a safety constraint that forbids this set

$$SC : \text{the controller must NOT provide } u \text{ when process state is } x \tag{21}$$

Therefore, for the eVTOL example above, the derived safety constraint would be: the FCC must inhibit the "Push Nose Down" transition logic when the radar altimeter reads below 50 ft, regardless of airspeed indicator status.

These constraints then become the requirements for the run-time assurance (RTA) monitors discussed in the next part. By systematically deriving these constraints during the design phase, STPA effectively identifies the safety behavioral logic of the system, minimizing the epistemic uncertainty related to how the system will behave before the aircraft ever takes off [132].

5.3.2. Executing Control with RTA

While STPA provides the static blueprint of safety constraints, run-time assurance (RTA) provides the dynamic mechanism to enforce them. As previously discussed, when we cannot predict the behavior of a component (like a neural network), we must control its output. RTA is the architectural embodiment of this philosophy.

1. The Necessity for Bridging the Traceability Gap Involved by AI/ML

The integration of Artificial Intelligence (AI) and Machine Learning (ML) into aviation has challenged the traditional certification standards like DO-178C. These standards rely on structural coverage and requirements traceability—the ability to trace every line of code back to a low-level requirement.

As noted in a survey on AI certification, there is a fundamental traceability gap that prevents one from tracing a specific neuron activation to a specific safety requirement [133]. Consequently, trying to certify a DNN using DO-178C is like trying to verify the reliability of a human pilot's brain by dissecting their neurons—it is the wrong level of abstraction. RTA circumvents this by admitting that the "Complex Function" (the AI) is inherently untrusted and essentially uncertifiable to high design assurance levels (DAL A/B). Instead of verifying the code, RTA verifies the architecture.

2. Design the Monitor–Switch Architecture for RTA

To bridge the gap between non-deterministic AI performance and deterministic safety requirements, the RTA framework is typically implemented using a monitor–switch architecture according to ASTM F3269-21 standard [127], as shown in Figure 4. This architecture wraps the non-deterministic component in a safety envelope composed of three distinct elements:

- Complex Function (CF): The high-performance, AI-driven controller (e.g., an adaptive dynamic inversion flight control algorithm). It has high uncertainty and is treated as untrusted.
- Recovery Function (RF): A simplified, low-performance controller (e.g., a classic PID loop). It is deterministic, physics-based, and formally verified to be trusted.
- Safety Monitor (SM): A logic block that observes the system state $x(t)$, according to the sensor input and the Complex Function's proposed action $u_{CF}(t)$.

- Switch: A logical gate to determine which output between CF and RF to choose as the final output. It is by default set to connect CF unless the safety monitor predicts that the system state would violate the safety boundary and triggers it to transit to RF.

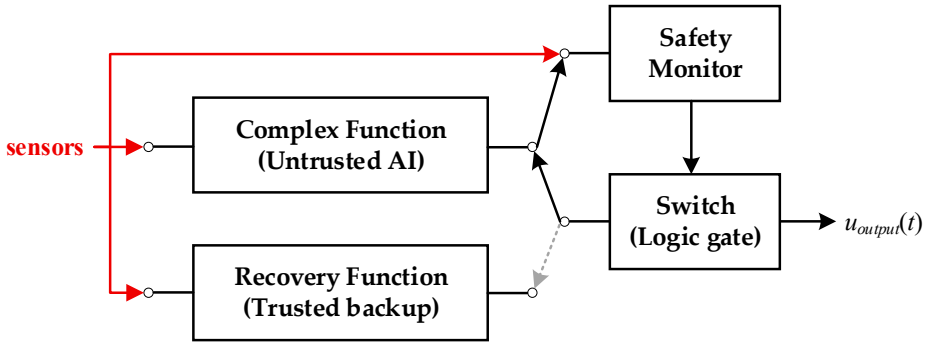

Figure 4. The RTA monitor–switch architecture.

The core logic of the monitor–switch architecture is to check if the system is approaching the boundary of the safety envelope Ω_{safe}. To do this, it often employs Forward Reachability Analysis. Let $R(x, \Delta t)$ be the set of all reachable states from state x within time Δt under the Recovery Function. The switching logic is defined as

$$u_{output}(t) = \begin{cases} u_{CF}(t), & \text{if } R(x(t) + f(x, u_{CF})\Delta t, T_{rec}) \subset \Omega_{safe}; \\ u_{RF}(t), & \text{otherwise.} \end{cases} \quad (22)$$

where T_{rec} is the time required for the Recovery Function to stabilize the system. The equation shows that the safety monitor is to determine "If I let the AI execute this command for one more step, will the backup controller still be able to save the aircraft?" If the answer is "No", the safety monitor immediately triggers and switches to Recovery Function.

- A Brief Case Study: Neural Network Flight Control

A study conducted by the Air Force Research Laboratory (AFRL) demonstrated this on an F-16 aircraft using a neural network for adaptive maneuvering. The unverified neural network provided superior tracking performance but occasionally exhibited oscillatory "bursts" due to over-fitting. The RTA wrapper, configured with a simple linear quadratic regulator as the Recovery Function, successfully intercepted 100% of the divergent behaviors, maintaining the aircraft within the structural G-load limits without suppressing the AI's performance during normal operation [134].

3. Closing the Evidence Chain: From Probabilistic to Deterministic

The profound innovation of RTA is how it transforms the safety argument for certification. In the traditional view, system safety is regarded as the combination of the components' reliability. However, in the resilience view enabled by RTA, the safety argument is decoupled as

$$Safety_{\text{system}} = Reliability_{\text{Monitor}} \times Reliability_{\text{Recovery}} \quad (23)$$

Hence, the engineers need to build a complete evidence chain for certification:

- The Recovery Function is verified to DAL A using traditional methods (safe by design).
- The safety monitor is verified to DAL A (simple logic, no complex math).
- The Switching Logic covers all STPA-identified hazardous states.

RTA is not just an academic concept but the enabling technology that will allow certified AI-driven aerospace systems to enter service in the coming decade.

5.3.3. Implementation Challenges for STPA and RTA

Although STPA and RTA offer a robust path toward system resilience, their proper implementation is subject to specific challenges that practitioners must recognize.

For STPA, it is a qualitative analysis method, not a quantitative one. Its effectiveness depends on the domain knowledge of the analysts. STPA cannot automatically identify hazards; it can only provide guidance for human experts to think systematically about system interactions. If the analysts do not obtain knowledge about system physics (e.g., unknown aerodynamic coupling effects), applying STPA may still miss critical UCAs.

For RTA, according to Equation (23), the safety guarantee of an RTA architecture relies entirely on the absolute reliability of the Recovery Function and the safety monitor. Therefore, the Recovery Function must be simple enough to be fully verified with DO-178C. If the Recovery Function itself is too complex or relies on the same flawed perception data as the AI, then the "common mode failure" will defeat the RTA. In addition, RTA only works well when a physical solution exists to recover the system, which means if the aircraft enters an irrecoverable state before the safety monitor triggers the Recovery Function, then RTA cannot prevent the accident. Therefore, the predefined safety envelope must be determined conservatively enough to allow sufficient time for the Recovery Function to initiate.

5.3.4. From Assurance to Evolution: The Closed-Loop Learning Mechanism

While RTA provides real-time resilience, the resilience paradigm further demands a systematic transition from static assurance to evolutionary safety. This is achieved through a continuous learning loop that embodies the core principles of Safety-II. In this framework, the RTA layer acts as a critical operational sensor: every instance of safety-filter intervention or switch to a backup controller is treated as a high-value data point rather than a mere system anomaly. As illustrated in Figure 5, these operational signals are analyzed to identify discrepancies between 'Work-as-Imagined' (design-time STPA models) and 'Work-as-Done' (actual system behavior under epistemic uncertainty). When the system operates in degraded modes or encounters unexpected environmental coupling, the RTA logs provide the empirical evidence needed to refine the simplified physics-based models and update the safety constraints. This feedback mechanism ensures that the system's uncertainty repertoire is not static but matures through operational experience, enabling continuous improvement of both the primary AI-based controllers and the deterministic assurance logic.

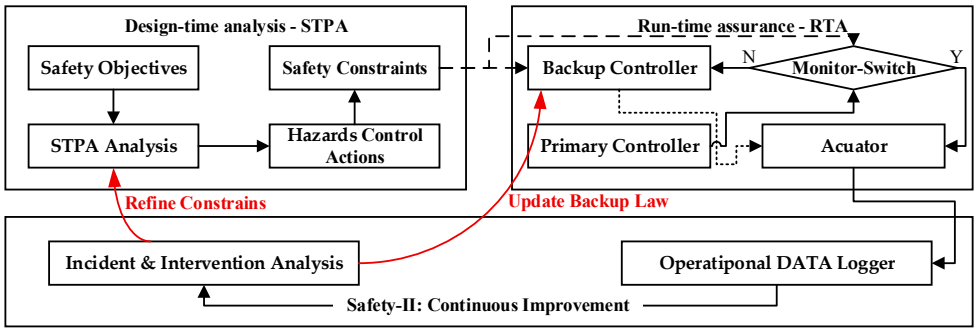

Figure 5. The closed-loop architecture of uncertainty control based on RTA.

5.4. The New Identity: The Engineer as a System Resilience Architect

The methodological evolution from Reliability Block Diagrams to Physics-of-Failure models, and then to RTA architectures, necessitates a parallel evolution in the practitioner's identity. The era of the specialist reliability engineer, who works in a silo to calculate MTBFs

and minimize failure rates, is drawing to a close. In the Resilience Era, this professional must evolve into a system resilience architect.

5.4.1. Synthesis of Disciplines: The T-Shaped Expert

The complexity of systems like UAM and autonomous defense platforms demands a synthesis of disciplines that were previously distinct. As highlighted in the INCOSE Systems Engineering Vision 2035, the future engineer must possess transdisciplinary skills. The system resilience architect is no longer just a statistician or a material physicist. They must now integrate:

- Control Theory: To understand stability, feedback loops, and STPA-based constraints.
- Software Engineering: To architect RTA wrappers and understand AI/ML behaviors.
- Systems Engineering: To manage the emergent interactions between hardware, software, and humans.

5.4.2. Role Definition: Designing the "Immune System"

The core mandate of the resilience architect differs fundamentally from that of the reliability engineer. The architect's job is to design the system's "immune system"—the adaptive architectures that sense, respond to, and recover from unforeseen disruptions. This is often quantified using the resilience triangle concept. If $Q(t)$ represents the system's quality of performance at time t, and a disruption occurs at t_e and ends at t_r, the "Loss of Resilience", denoted as L_{Res}, that the architect must minimize is defined as

$$L_{\text{Res}} = \int_{t_e}^{t_r} [1 - Q(t)]dt \tag{24}$$

By doing so, we can expand the expression of the capability of a system, as the dynamic redundant or recovery mechanisms are designed into the system architecture. For instance, in the event of a sensor failure, a "Reliable" system might simply disengage (dropping $Q(t)$ to 0). Whereas a "Resilient" system might maintain $Q(t) = 90\%$, if the missing sensor data can be constructed from redundant sensors.

5.4.3. Conclusion of the New Era: Enveloping, Not Replacing

It is crucial to understand that the resilience paradigm does not discard the previous eras. It is not a revolution of destruction, but of envelopment. That is to say:

- We still need statistics to model the stochastic failure of the hardware components used in the system.
- We still need physics-of-failure to design the sensors and actuators that constitute the physical plant.
- We still need prognostics to feed accurate state data to manage the uncertainty dynamically.

The Resilience Era wraps these foundational layers in a higher-level framework of uncertainty control. The system resilience architect treats component reliability as a resource to be managed, prognostics as intelligence to be consumed, and control architecture as the mechanism to deliver mission success.

As we move deeper into the 2020s, the systems we build—from urban air taxis to deep-space autonomous probes—will face uncertainties we cannot yet imagine. By embracing the identity of the resilience architect and mastering the tools of uncertainty control, the engineering profession ensures it remains the guardian of safety in this brave new world of complexity [135].

6. Conclusions

The evolution of reliability engineering is a mirror reflecting the increasing complexity of the technological systems we build. As reviewed in this paper, the discipline has undergone a profound transformation through three historical phases, each responding to the dominant uncertainty of its time. The Statistical Era established the mathematical foundations for managing the aleatory uncertainty of mass-produced hardware, treating the system as a "black box" governed by probability distributions. The Physics-of-Failure Era opened this box, providing the causal understanding of degradation mechanisms needed to "design in" reliability against physical stress. The Prognostics Era further advanced this by introducing the dimension of time, leveraging sensor data and digital twins to transform static reliability estimates into dynamic, real-time health management. However, the nature of safety-critical systems—defined by software interconnectivity, high autonomy, and "black-box" AI components—has pushed these failure-centric paradigms to their limits. A primary challenge is no longer just the stochastic breakdown of a part, but the emergent, unsafe interaction of functional components. Because these systems are beyond the sum of their parts, safety can no longer be assured solely by verifying the parts; it must be architected at the system level.

This paper argues that the future of the profession lies in the Resilience Era, a paradigm shift that redefines the reliability engineering objective from preventing failure (Safety-I) to ensuring mission success under uncertainty (Safety-II). To bridge the gap between this high-level goal and engineering practice, this study explicitly posits that UC serves as the operational mechanism to achieve system resilience. By implementing RTA architecture, engineers can create a deterministic safety envelope that acts as a definitive backstop for the system. When non-deterministic components (like an AI model) encounter OOD scenarios and fail to predict correctly, the UC mechanism intercepts the unsafe command and enforces a verified recovery strategy. Thus, the system remains resilient not because it is perfect at prediction, but because it is architected to contain the consequences of prediction failure. Ultimately, the resilience paradigm does not discard the tools of the past but envelopes them. The reliability engineer is no longer just to calculate failure rates, understand degradation, or forecast RUL, but to synthesize these inputs into a hierarchical control framework that renders the system immune to any unknown situations.

Funding: This research was funded by Aviation Industry Corporation of China with the project entitled "Safety and Reliability Design and Assessment Techniques for Low-altitude Aircraft based on Modern Safety Theory". The APC was funded by China Aero-Polytechnology Establishment.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Acknowledgments: During the preparation of this work, the authors used Gemini 2.5 in order to improve the language and readability of the manuscript. The authors have reviewed and edited the output and take full responsibility for the content of this publication.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AFRL	Air Force Research Laboratory
AI	Artificial Intelligence
ARP	Aerospace Recommended Practice
ASIC	Application-Specific Integrated Circuit
ASTM	American Society for Testing and Materials

CALCE	Center for Advanced Life Cycle Engineering
CBF	Control Barrier Function
CBM	Condition-Based Maintenance
CCA	Common Cause Analysis
CCF	Common Cause Failure
CEA	Cascading Effects Analysis
CF	Complex Function
CMS	Central Maintenance System
CNN	Convolutional Neural Network
DAL	Design Assurance Level
DEP	Distributed Electric Propulsion
DNN	Deep Neural Network
EASA	European Union Aviation Safety Agency
eVTOL	electric Vertical Take-Off and Landing
FMEA	Failure Modes and Effects Analysis
FTA	Fault Tree Analysis
GRU	Gated Recurrent Unit
HUMS	Health and Usage Monitoring Systems
IMA	Integrated Modular Avionics
IRU	Inertial Reference Unit
KF	Kalman Filter
LSTM	Long Short-Term Memory
MCAS	Maneuvering Characteristics Augmentation System
MIL-HDBK	Military Handbook
ML	Machine Learning
MOSFET	Metal-Oxide-Semiconductor Field-Effect Transistor
MPC	Model Predictive Control
MSE	Mean Squared Error
MTBF	Mean Time Between Failures
MTTF	Mean Time To Failure
NASA	National Aeronautics and Space Administration
NTSB	National Transportation Safety Board
OOD	Out-of-Distribution
PDE	Partial Differential Equation
PDF	Probability Density Function
PF	Particle Filter
PHM	Prognostics and Health Management
PINN	Physics-Informed Neural Network
PoF	Physics-of-Failure
PRA	Probabilistic Risk Assessment
PSA	Probabilistic Safety Assessment
RBD	Reliability Block Diagram
RF	Recovery Function
RNN	Recurrent Neural Network
RPN	Risk Priority Number
RTA	Run-Time Assurance
RUL	Remaining Useful Life
SAE	Society of Automotive Engineers
SEI	Solid-Electrolyte Interphase
SM	Safety Monitor
SPC	Statistical Process Control
STAMP	Systems-Theoretic Accident Model and Processes
STPA	Systems-Theoretic Process Analysis

UAM	Urban Air Mobility
UC	Uncertainty Control
UCA	Unsafe Control Action
UQ	Uncertainty Quantification

References

1. Faruk, M.J.H.; Miner, P.; Coughlan, R.; Masum, M.; Shahriar, H.; Clincy, V.; Cetinkaya, C. Smart Connected Aircraft: Towards Security, Privacy, and Ethical Hacking. In Proceedings of the 2021 14th International Conference on Security of Information and Networks (SIN), Edinburgh, UK, 15–17 December 2021; Volume 1, pp. 1–5. [CrossRef]
2. Koopman, P.; Wagner, M. Autonomous Vehicle Safety: An Interdisciplinary Challenge. *IEEE Intell. Transp. Syst. Mag.* **2017**, *9*, 90–96. [CrossRef]
3. Brelje, B.J.; Martins, J.R.R.A. Electric, Hybrid, and Turboelectric Fixed-Wing Aircraft: A Review of Concepts, Models, and Design Approaches. *Prog. Aerosp. Sci.* **2019**, *104*, 1–19. [CrossRef]
4. Kabzan, J.; Hewing, L.; Liniger, A.; Zeilinger, M.N. Learning-Based Model Predictive Control for Autonomous Racing. *IEEE Robot. Autom. Lett.* **2019**, *4*, 3363–3370. [CrossRef]
5. Liu, X.; Yuan, Z.; Gao, Z.; Zhang, W. Reinforcement Learning-Based Fault-Tolerant Control for Quadrotor UAVs Under Actuator Fault. *IEEE Trans. Ind. Inform.* **2024**, *20*, 13926–13935. [CrossRef]
6. Gaska, T.; Watkin, C.; Chen, Y. Integrated Modular Avionics—Past, Present, and Future. *IEEE Aerosp. Electron. Syst. Mag.* **2015**, *30*, 12–23. [CrossRef]
7. Zhao, C.; Dong, L.; Li, H.; Wang, P. Safety Assessment of the Reconfigurable Integrated Modular Avionics Based on STPA. *Int. J. Aerosp. Eng.* **2021**, *2021*, 8875872. [CrossRef]
8. Wise, K.A.; Lavretsky, E.; Hovakimyan, N. Adaptive Control of Flight: Theory, Applications, and Open Problems. In Proceedings of the 2006 American Control Conference, Minneapolis, MN, USA, 14–16 June 2006; p. 6.
9. Soukkou, Y.; Tadjine, M.; Zhu, Q.M.; Nibouche, M. Robust Adaptive Sliding Mode Control Strategy of Uncertain Nonlinear Systems. *Proc. Inst. Mech. Eng. Part G J. Aerosp. Eng.* **2023**, *237*, 62–74. [CrossRef]
10. Leveson, N. *Safety III: A Systems Approach to Safety and Resilience*; Mit Engineering Systems Lab: Cambridge, MA, USA, 2020.
11. Patriarca, R.; Chatzimichailidou, M.; Karanikas, N.; Gravio, G.D. The Past and Present of System-Theoretic Accident Model and Processes (STAMP) and Its Associated Techniques: A Scoping Review. *Saf. Sci.* **2022**, *146*, 105566. [CrossRef]
12. Endsley, M.R. Autonomous Driving Systems: A Preliminary Naturalistic Study of the Tesla Model S. *J. Cogn. Eng. Decis. Mak.* **2017**, *11*, 225–238. [CrossRef]
13. Banks, V.A.; Plant, K.L.; Stanton, N.A. Driver Error or Designer Error: Using the Perceptual Cycle Model to Explore the Circumstances Surrounding the Fatal Tesla Crash on 7th May 2016. *Saf. Sci.* **2018**, *108*, 278–285. [CrossRef]
14. Dekker, S. *The Field Guide to Understanding 'Human Error'*, 3rd ed.; CRC Press: Boca Raton, FL, USA, 2014.
15. Hollnagel, E. *Safety-I and Safety-II: The Past and Future of Safety Management*; CRC Press: Boca Raton, FL, USA, 2014.
16. Carlson, C.S. *Effective FMEAs: Achieving Safe, Reliable, and Economical Products and Processes Using Failure Mode and Effects Analysis*; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 2012.
17. National Transportation Safety Board. *Response to Final Aircraft Accident Investigation Report Ethiopian Airlines Flight 302 Boeing 737-8 MAX, ET-AVJ Ejere, Ethiopia 10 March 2019*; National Transportation Safety Board: Washington, DC, USA, 2019.
18. Sadeqi, O. *Applying Stpa for Safety Analysis of Autonomous Vehicles*; Mälardalen University: Eskilstuna, Sweden, 2024.
19. JATR Team. *Boeing 737 MAX Flight Control System*; JATR Team: Washington, DC, USA, 2019.
20. National Transportation Safety Board. *Collision Between Vehicle Controlled by Developmental Automated Driving System and Pedestrian Tempe, Arizona 18 March 2018*; National Transportation Safety Board: Washington, DC, USA, 2019.
21. National Highway Traffic Safety Administration (NHTSA), Department of Transportation. *Federal Motor Vehicle Safety Standards; Automatic Emergency Braking Systems for Light Vehicles*; National Highway Traffic Safety Administration (NHTSA), Department of Transportation: Washington, DC, USA, 2024.
22. Helton, J.C.; Johnson, J.D.; Oberkampf, W.L. An Exploration of Alternative Approaches to the Representation of Uncertainty in Model Predictions. *Reliab. Eng. Syst. Saf.* **2004**, *85*, 39–71. [CrossRef]
23. Xiong, F.F.; Li, Z.X.; Liu, Y.; Xiahou, T.F. A review of characterization methods for parameter uncertainty in engineering design based on numerical simulation. *Acta Aeronaut. Astronaut. Sin.* **2023**, *44*, 028611. (In Chinese) [CrossRef]
24. Kersting, S.; Kohler, M. Uncertainty Quantification in Case of Imperfect Models: A Review. *arXiv* **2020**, arXiv:2012.09449. [CrossRef]
25. Roy, C.J.; Oberkampf, W.L. A Comprehensive Framework for Verification, Validation, and Uncertainty Quantification in Scientific Computing. *Comput. Methods Appl. Mech. Eng.* **2011**, *200*, 2131–2144. [CrossRef]
26. Gawlikowski, J.; Tassi, C.R.N.; Ali, M.; Lee, J.; Humt, M.; Feng, J.; Kruspe, A.; Triebel, R.; Jung, P.; Roscher, R.; et al. A Survey of Uncertainty in Deep Neural Networks. *Artif. Intell. Rev.* **2023**, *56*, 1513–1589. [CrossRef]

27. Neto, A.V.S.; Camargo, J.B.; Almeida, J.R.; Cugnasca, P.S. Safety Assurance of Artificial Intelligence-Based Systems: A Systematic Literature Review on the State of the Art and Guidelines for Future Work. *IEEE Access* **2022**, *10*, 130733–130770. [CrossRef]
28. Shi, Y.; Wei, P.; Feng, K.; Feng, D.-C.; Beer, M. A Survey on Machine Learning Approaches for Uncertainty Quantification of Engineering Systems. *Mach. Learn. Comput. Sci. Eng.* **2025**, *1*, 11. [CrossRef]
29. *SAE 4754B*; Guidelines for Development of Civil Aircraft and Systems. SAE International: Warrendale, PA, USA, 2023.
30. *SAE 4761A*; Guidelines for Conducting the Safety Assessment Process on Civil Aircraft, Systems, and Equipment. SAE International: Warrendale, PA, USA, 2023.
31. Sainani, K.L. Reliability statistics. *PM&R* **2017**, *9*, 622–628. [CrossRef]
32. Dong, Y.; Huang, W.; Bharti, V.; Cox, V.; Banks, A.; Wang, S.; Zhao, X.; Schewe, S.; Huang, X. Reliability Assessment and Safety Arguments for Machine Learning Components in System Assurance. *ACM Trans. Embed. Comput. Syst.* **2023**, *22*, 1–48. [CrossRef]
33. Chen, S.; Sun, Y.; Li, D.; Wang, Q.; Hao, Q.; Sifakis, J. Runtime Safety Assurance for Learning-Enabled Control of Autonomous Driving Vehicles. In *Proceedings of the 2022 International Conference on Robotics and Automation (ICRA), Philadelphia, PA, USA, 23–27 May 2022*; IEEE: New York, NY, USA, 2022; pp. 8978–8984.
34. Meeker, W.Q.; Escobar, L.A.; Pascual, F.G. *Statistical Methods for Reliability Data*; John Wiley & Sons: Hoboken, NJ, USA, 2021; ISBN 1-118-11545-7.
35. Foucher, B.; Boullie, J.; Meslet, B.; Das, D. A Review of Reliability Prediction Methods for Electronic Devices. *Microelectron. Reliab.* **2002**, *42*, 1155–1162. [CrossRef]
36. Zio, E. Prognostics and health management: A review from the perspectives of design, development and decision. *Reliab. Eng. Syst. Saf.* **2022**, *217*, 108063. [CrossRef]
37. O'Connor, P.D.T.; Kleyner, A.V. *Practical Reliability Engineering*, 5th ed.; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 2012.
38. Lai, C.-D.; Xie, M.; Murthy, D.N.P. Bathtub Shaped Failure Rate Life Distributions. In *Stochastic Ageing and Dependence for Reliability*; Springer Nature: London, UK, 2006; pp. 71–107.
39. *Military Handbook Reliability Prediction of Electronic Equipment*; MIL-HDBK-217F; Department of Defense: Washington, DC, USA, 1991.
40. Luko, S.N. A Review of the Weibull Distribution and Selected Engineering Applications. *SAE Trans.* **1999**, *108*, 398–412.
41. Wais, P. Two and Three-Parameter Weibull Distribution in Available Wind Power Analysis. *Renew. Energy* **2017**, *103*, 15–29. [CrossRef]
42. Ditlevsen, O.; Madsen, H.O. *Structural Reliability Methods*; Wiley: New York, NY, USA, 1996; Volume 178.
43. Choi, S.-K.; Canfield, R.A.; Grandhi, R.V. *Reliability-Based Structural Design*; Springer: Berlin/Heidelberg, Germany, 2007.
44. Li, S.; Chen, Z.; Liu, Q.; Shi, W.; Li, K. Modeling and Analysis of Performance Degradation Data for Reliability Assessment: A Review. *IEEE Access* **2020**, *8*, 74648–74678. [CrossRef]
45. Zhao, Y.; Yang, B.; Peng, J. Reconstruction of Probabilistic S-N Curves under Fatigue Life Following Lognormal Distribution with given Confidence. *Appl. Math. Mech.* **2007**, *28*, 455–460. [CrossRef]
46. Singpurwalla, N.D. *Reliability and Risk: A Bayesian Perspective*; John Wiley & Sons: Hoboken, NJ, USA, 2006; ISBN 0-470-06033-6.
47. Lee, Y.-L.; Makam, S.; McKelvey, S.; Lu, M.-W. Durability Reliability Demonstration Test Methods. *Procedia Eng.* **2015**, *133*, 31–59. [CrossRef]
48. Martz, H.F., Jr.; Waller, R.A. A Bayesian Zero-Failure (BAZE) Reliability Demonstration Testing Procedure. *J. Qual. Technol.* **1979**, *11*, 128–138. [CrossRef]
49. Tasias, K.A.; Alevizakos, V. Cumulative Sum Control Charts for Monitoring Zero-inflated COM-Poisson Processes: CUSUM Charts for ZICMP Distribution. *Qual. Reliab. Eng. Int.* **2024**, *40*, 2891–2903. [CrossRef]
50. Luo, F.; Hu, L.; Wang, Y.; Yu, X. Statistical Inference of Reliability for a K-out-of-N: G System with Switching Failure under Poisson Shocks. *Stat. Theory Relat. Fields* **2024**, *8*, 195–210. [CrossRef]
51. Coit, D.W.; Jin, T. Gamma Distribution Parameter Estimation for Field Reliability Data with Missing Failure Times. *Iie Trans.* **2000**, *32*, 1161–1166.
52. Rausand, M.; Hoyland, A. *System Reliability Theory: Models, Statistical Methods, and Applications*; John Wiley & Sons: Hoboken, NJ, USA, 2003; Volume 396, ISBN 0-471-47133-X.
53. Khan, Z.; Al-Bossly, A.; Almazah, M.M.; Alduais, F.S. On Statistical Development of Neutrosophic Gamma Distribution with Applications to Complex Data Analysis. *Complexity* **2021**, *2021*, 3701236. [CrossRef]
54. Justin, C.; Patel, S.; Bouchard, E.D.; Gladin, J.; Verberne, J.; Li, E.; Ozcan, M.; Rajaram, D.; Mavris, D.; D'Arpino, M. *Reliability and Safety Assessment of Urban Air Mobility Concept Vehicles*; National Aeronautics and Space Administration (NASA) Ames Research Center: Moffett Field, CA, USA, 2021.
55. Cheng, L.; Wan, Y.; Zhou, Y.; Gao, D.W. Operational Reliability Modeling and Assessment of Battery Energy Storage Based on Lithium-Ion Battery Lifetime Degradation. *J. Mod. Power Syst. Clean Energy* **2021**, *10*, 1738–1749. [CrossRef]
56. Baladeh, A.E.; Taghipour, S. Reliability Optimization of Dynamic K-out-of-n Systems with Competing Failure Modes. *Reliab. Eng. Syst. Saf.* **2022**, *227*, 108734. [CrossRef]

57. Eryılmaz, S. Reliability Properties of Consecutive K-out-of-n Systems of Arbitrarily Dependent Components. *Reliab. Eng. Syst. Saf.* **2009**, *94*, 350–356. [CrossRef]
58. Lin, C.; Zeng, Z.; Zhou, Y.; Xu, M.; Ren, Z. A Lower Bound of Reliability Calculating Method for Lattice System with Non-Homogeneous Components. *Reliab. Eng. Syst. Saf.* **2019**, *188*, 36–46. [CrossRef]
59. Jia, H.; Peng, R.; Yang, L.; Wu, T.; Liu, D.; Li, Y. Reliability Evaluation of Demand-Based Warm Standby Systems with Capacity Storage. *Reliab. Eng. Syst. Saf.* **2022**, *218*, 108132. [CrossRef]
60. Kumar, A.; Garg, R.; Barak, M.S. Reliability Measures of a Cold Standby System Subject to Refreshment. *Int. J. Syst. Assur. Eng. Manag.* **2023**, *14*, 147–155. [CrossRef]
61. Frangopol, D.M.; Maute, K. Reliability-Based Optimization of Civil and Aerospace Structural Systems. In *Engineering Design Reliability Handbook*; CRC Press: Boca Raton, FL, USA, 2004; pp. 559–590.
62. Ke, H.-Y. A Bayesian/Classical Approach to Reliability Demonstration. *Qual. Eng.* **2000**, *12*, 365–370. [CrossRef]
63. Xiong, J.; Shenoi, R.A.; Gao, Z. Small Sample Theory for Reliability Design. *J. Strain Anal. Eng. Des.* **2002**, *37*, 87–92. [CrossRef]
64. Mosleh, A. Common Cause Failures: An Analysis Methodology and Examples. *Reliab. Eng. Syst. Saf.* **1991**, *34*, 249–292. [CrossRef]
65. Pecht, M.G. Prognostics and Health Management. In *Solid State Lighting Reliability*; Van Driel, W., Fan, X., Eds.; Solid State Lighting Technology and Application Series; Springer: New York, NY, USA, 2013; Volume 1. [CrossRef]
66. Pecht, M. Prognostics and Health Management of Electronics. In *Encyclopedia of Structural Health Monitoring*; John Wiley & Sons: Hoboken, NJ, USA, 2009.
67. Varde, P.V. Physics-of-Failure Based Approach for Predicting Life and Reliability of Electronics Components. *Barc Newsletter* **2010**, *313*, 38–46.
68. Hendricks, C.; George, E.; Osterman, M.; Pecht, M. Physics-of-Failure (PoF) Methodology for Electronic Reliability. In *Reliability Characterisation of Electrical and Electronic Systems*; Swingler, J., Ed.; Woodhead Publishing: Oxford, UK, 2015; pp. 27–42, ISBN 978-1-78242-221-1.
69. White, M.; Bernstein, J.B. *Microelectronics Reliability: Physics-of-Failure Based Modeling and Lifetime Evaluation*; Jet Propulsion Laboratory, National Aeronautics and Space Administration: Pasadena, CA, USA, 2008.
70. Temsamani, A.B.; Kauffmann, S.; Helsen, S.; Gaens, T.; Driesen, V. Physics-of-Failure (PoF) methodology for qualification and lifetime assessment of supercapacitors for industrial applications. *Microelectron. Reliab.* **2018**, *88*, 54–60. [CrossRef]
71. Chai, M.; Hou, X.; Zhang, Z.; Duan, Q. Identification and prediction of fatigue crack growth under different stress ratios using acoustic emission data. *Int. J. Fatigue* **2022**, *160*, 106860. [CrossRef]
72. Lindsey, N.J. *NASA Methodology for Physics of Failure-Based Reliability Assessments Handbook*; Goddard Space Flight Center, National Aeronautics and Space Administration: Greenbelt, MD, USA, 2024.
73. Grandt, A.F., Jr. *Fundamentals of Structural Integrity: Damage Tolerant Design and Nondestructive Evaluation*; John Wiley & Sons: Hoboken, NJ, USA, 2003; ISBN 0-471-21459-0.
74. Kedir, Y.A.; Lemu, H.G. Prediction of Fatigue Crack Initiation under Variable Amplitude Loading: Literature Review. *Metals* **2023**, *13*, 487. [CrossRef]
75. Pierce, D.G.; Brusius, P.G. Electromigration: A Review. *Microelectron. Reliab.* **1997**, *37*, 1053–1072. [CrossRef]
76. Pecht, M.; Gu, J. Physics-of-failure-based prognostics for electronic products. *Trans. Inst. Meas. Control* **2009**, *31*, 309–322. [CrossRef]
77. Yang, D. Physics-of-Failure-Based Prognostics and Health Management for Electronic Products. In Proceedings of the 2014 15th International Conference on Electronic Packaging Technology, Chengdu, China, 12–15 August 2014; pp. 1215–1218.
78. Stathis, J.H.; Zafar, S. The Negative Bias Temperature Instability in MOS Devices: A Review. *Microelectron. Reliab.* **2006**, *46*, 270–286. [CrossRef]
79. Schroder, D.K. Negative Bias Temperature Instability: What Do We Understand? *Microelectron. Reliab.* **2007**, *47*, 841–852. [CrossRef]
80. Bender, E.; Bernstein, J.B.; Boning, D.S. Modern Trends in Microelectronics Packaging Reliability Testing. *Micromachines* **2024**, *15*, 398. [CrossRef]
81. Lang, F.; Zhou, Z.; Liu, J.; Cui, M.; Zhang, Z. Review on the Impact of Marine Environment on the Reliability of Electronic Packaging Materials. *Front. Mater.* **2025**, *12*, 1584349. [CrossRef]
82. Zhu, S.P.; Huang, H.Z.; Peng, W.; Wang, H.K.; Mahadevan, S. Probabilistic physics of failure-based framework for fatigue life prediction of aircraft gas turbine discs under uncertainty. *Reliab. Eng. Syst. Saf.* **2016**, *146*, 1–12. [CrossRef]
83. Dai, Y.; Panahi, A. Thermal Runaway Process in Lithium-Ion Batteries: A Review. *Next Energy* **2025**, *6*, 100186. [CrossRef]
84. Ramesh, T.; Janis, V. *Modeling Damage, Fatigue and Failure of Composite Materials*, 2nd ed.; Elsevier: Amsterdam, The Netherlands, 2023.
85. Shrivastava, P. Application of FMEA in Developing Design and Reliability Verification Plan. In Proceedings of the 2023 Annual Reliability and Maintainability Symposium (RAMS), Orlando, FL, USA, 23–26 January 2023; pp. 1–6.
86. Sharma, K.D.; Srivastava, S. Failure Mode and Effect Analysis (FMEA) Implementation: A Literature Review. *J. Adv. Res. Aeronaut. Space Sci.* **2018**, *5*, 1–17.

87. Vesely, W.E.; Goldberg, F.F.; Roberts, N.H.; Haasl, D.F. *Fault Tree Handbook*; U.S. Nuclear Regulatory Commission: Rockville, MD, USA, 1981.
88. Ejaz, M.R.; Chikonde, M. *Stpa for Autonomous Vehicle Safety in Traffic Systems*; Chalmers University of technology: Gothenburg, Sweden, 2022.
89. Fan, J.; Yung, K.C.; Pecht, M. Physics-of-Failure-Based Prognostics and Health Management for High-Power White Light-Emitting Diode Lighting. *IEEE Trans. Device Mater. Reliab.* **2011**, *11*, 407–416. [CrossRef]
90. Jin, G.; Matthews, D.; Fan, Y.; Liu, Q. Physics of failure-based degradation modeling and lifetime prediction of the momentum wheel in a dynamic covariate environment. *Eng. Fail. Anal.* **2013**, *28*, 222–240. [CrossRef]
91. Marliere, T.A.; Cesar, C.d.A.C.; Hirata, C.M. Extending the STPA to Model the Control Structure with Finite State Machine. *J. Saf. Sci. Resil.* **2025**, *6*, 100214. [CrossRef]
92. Holley, S.; Miller, M. Cognitive Processing Disruptions Affecting Flight Deck Performance: Implications for Cognitive Resilience. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*; SAGE Publications Sage: Los Angeles, CA, USA, 2023; Volume 67, pp. 2101–2106.
93. Zio, E. Prognostics and Health Management (PHM): Where Are We and Where Do We (Need to) Go in Theory and Practice. *Reliab. Eng. Syst. Saf.* **2022**, *218*, 108119. [CrossRef]
94. Yan, R.; Zhou, Z.; Shang, Z.; Wang, Z.; Hu, C.; Li, Y.; Yang, Y.; Chen, X.; Gao, R.X. Knowledge Driven Machine Learning towards Interpretable Intelligent Prognostics and Health Management: Review and Case Study. *Chin. J. Mech. Eng.* **2025**, *38*, 5. [CrossRef]
95. Elattar, H.M.; Elminir, H.K.; Riad, A.M. Prognostics: A Literature Review. *Complex Intell. Syst.* **2016**, *2*, 125–154. [CrossRef]
96. Lindsey, N.J.; Dawson, J.; Sheldon, D.; Sindjui, L.N.; DiVenti, A. NASA Physics of Failure (PoF) for Reliability. In Proceedings of the Probabilistic Safety Assessment & Management (PSAM-16) Conference, Honolulu, HI, USA, 26 June–1 July 2022.
97. Gu, J.; Pecht, M. Prognostics and health management using physics-of-failure. In Proceedings of the 2008 Annual Reliability and Maintainability Symposium, Las Vegas, NV, USA, 28–31 January 2008; pp. 481–487. [CrossRef]
98. Giurgiutiu, V. *Structural Health Monitoring of Aerospace Composites*; Academic Press: San Diego, CA, USA, 2015.
99. Guillén, A.J.; Crespo, A.; Macchi, M.; Gómez, J. On the Role of Prognostics and Health Management in Advanced Maintenance Systems. *Prod. Plan. Control* **2016**, *27*, 991–1004. [CrossRef]
100. An, D.; Choi, J.H.; Kim, N.H. Options for Prognostics Methods: A Review of Data-Driven and Physics-Based Prognostics. In Proceedings of the 54th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Boston, MA, USA, 8–11 April 2013; p. 1940.
101. Feng, J.; Cai, F.; Li, H.; Huang, K.; Yin, H. A Data-Driven Prediction Model for the Remaining Useful Life Prediction of Lithium-Ion Batteries. *Process Saf. Environ. Prot.* **2023**, *180*, 601–615. [CrossRef]
102. Li, W.; Chen, J.; Chen, S.; Li, P.; Zhang, B.; Wang, M.; Yang, M.; Wang, J.; Zhou, D.; Yun, J. A Comprehensive Review of Artificial Intelligence-Based Algorithms for Predicting the Remaining Useful Life of Equipment. *Sensors* **2025**, *25*, 4481. [CrossRef] [PubMed]
103. Zhang, L.; Lin, J.; Liu, B.; Zhang, Z.; Yan, X.; Wei, M. A Review on Deep Learning Applications in Prognostics and Health Management. *IEEE Access* **2019**, *7*, 162415–162438. [CrossRef]
104. Kulkarni, C.S. Hybrid Approaches to Systems Health Management and Prognostics. In Proceedings of the Workshop on "Prognostics and Health Management", Virtual, 29 October 2021.
105. Polverino, L.; Abbate, R.; Manco, P.; Perfetto, D.; Caputo, F.; Macchiaroli, R.; Caterino, M. Machine Learning for Prognostics and Health Management of Industrial Mechanical Systems and Equipment: A Systematic Literature Review. *Int. J. Eng. Bus. Manag.* **2023**, *15*, 18479790231186848. [CrossRef]
106. Kim, S.; Seo, Y.-H.; Park, J. Transformer-Based Novel Framework for Remaining Useful Life Prediction of Lubricant in Operational Rolling Bearings. *Reliab. Eng. Syst. Saf.* **2024**, *251*, 110377. [CrossRef]
107. Wang, R.; Dong, E.; Cheng, Z.; Liu, Z.; Jia, X. Transformer-Based Intelligent Fault Diagnosis Methods of Mechanical Equipment: A Survey. *Open Phys.* **2024**, *22*, 20240015. [CrossRef]
108. Farbiz, F.; Habibullah, M.S.; Hamadicharef, B.; Maszczyk, T.; Aggarwal, S. Knowledge-embedded machine learning and its applications in smart manufacturing. *J. Intell. Manuf.* **2023**, *34*, 2889–2906. [CrossRef]
109. Artelt, M.; Weiß, M.; Dittler, D.; Goersch, Y.; Jazdi, N.; Weyrich, M. Hybrid Approaches and Datasets for Remaining Useful Life Prediction: A Review. *Procedia CIRP* **2024**, *130*, 294–300. [CrossRef]
110. Ferreira, C.; Gonçalves, G. Remaining Useful Life Prediction and Challenges: A Literature Review on the Use of Machine Learning Methods. *J. Manuf. Syst.* **2022**, *63*, 550–562. [CrossRef]
111. Cao, H.; Xiao, W.; Sun, J.; Gan, M.-G.; Wang, G. A Hybrid Data- and Model-Driven Learning Framework for Remaining Useful Life Prognostics. *Eng. Appl. Artif. Intell.* **2024**, *135*, 108557. [CrossRef]
112. Lixin, E.; Wang, J.; Yang, R.; Wang, C.; Li, H.; Xiong, R. A physics-informed neural network-based method for predicting degradation trajectories and remaining useful life of supercapacitors. *Green Energy Intell. Transp.* **2025**, *4*, 100291.

113. Li, H.; Zhang, Z.; Li, T.; Si, X. A Review on Physics-Informed Data-Driven Remaining Useful Life Prediction: Challenges and Opportunities. *Mech. Syst. Signal Process.* **2024**, *209*, 111120. [CrossRef]
114. Ahwiadi, M.; Wang, W. An AI-Driven Particle Filter Technology for Battery System State Estimation and RUL Prediction. *Batteries* **2024**, *10*, 437. [CrossRef]
115. Cui, L.; Wang, X.; Wang, H.; Ma, J. Research on Remaining Useful Life Prediction of Rolling Element Bearings Based on Time-Varying Kalman Filter. *IEEE Trans. Instrum. Meas.* **2019**, *69*, 2858–2867. [CrossRef]
116. Duan, B.; Zhang, Q.; Geng, F.; Zhang, C. Remaining Useful Life Prediction of Lithium-ion Battery Based on Extended Kalman Particle Filter. *Int. J. Energy Res.* **2020**, *44*, 1724–1734. [CrossRef]
117. Wu, T.; Zhao, T.; Xu, S. Prediction of Remaining Useful Life of the Lithium-Ion Battery Based on Improved Particle Filtering. *Front. Energy Res.* **2022**, *10*, 863285. [CrossRef]
118. Kim, S.; Choi, J.-H.; Kim, N.H. Data-Driven Prognostics with Low-Fidelity Physical Information for Digital Twin: Physics-Informed Neural Network. *Struct. Multidiscip. Optim.* **2022**, *65*, 255. [CrossRef]
119. Wen, P.; Ye, Z.-S.; Li, Y.; Chen, S.; Xie, P.; Zhao, S. Physics-Informed Neural Networks for Prognostics and Health Management of Lithium-Ion Batteries. *IEEE Trans. Intell. Veh.* **2023**, *9*, 2276–2289. [CrossRef]
120. Beaulieu, M.H.d.; Jha, M.S.; Garnier, H.; Cerbah, F. Remaining Useful Life Prediction Based on Physics-Informed Data Augmentation. *Reliab. Eng. Syst. Saf.* **2024**, *252*, 110451. [CrossRef]
121. Carreño, V.A. *ATM-X Urban Air Mobility: Assistive Detect and Avoid for UAM Operations Safety Evaluation Metrics*; NASA: Compass Engineering: San Juan, Puerto Rico, 2023.
122. Erik, H. *Synesis: The Unification of Productivity, Quality, Safety and Reliability*, 1st ed.; Routledge: Abingdon, UK, 2020; ISBN 978-0-367-48149-0.
123. Endsley, M.R. Situation Awareness in Future Autonomous Vehicles: Beware of the Unexpected. In *Proceedings of the 20th Congress of the International Ergonomics Association (IEA 2018)*; Bagnara, S., Tartaglia, R., Albolino, S., Alexander, T., Fujita, Y., Eds.; Springer International Publishing: Cham, Switzerland, 2019; pp. 303–309.
124. Leveson, N.G.; Thomas, J.P. *Stpa Handbook*; MIT Partnership for Systems Approaches to Safety and Security (PSASS): Cambridge, MA, USA, 2018.
125. Ames, A.D.; Coogan, S.; Egerstedt, M.; Notomista, G.; Sreenath, K.; Tabuada, P. Control Barrier Functions: Theory and Applications. In Proceedings of the 2019 18th European Control Conference (ECC), Naples, Italy, 25–28 June 2019; pp. 3420–3431.
126. Cheng, R.; Orosz, G.; Murray, R.M.; Burdick, J.W. End-to-End Safe Reinforcement Learning through Barrier Functions for Safety-Critical Continuous Control Tasks. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019.
127. *ASTM F3269*; Standard Practice for Methods to Safely Bound Flight Behavior of Unmanned Aircraft Systems Containing Complex Functions. ASTM International: West Conshohocken, PA, USA, 2021.
128. EASA. *Artificial Intelligence Roadmap 2.0: A Human-Centric Approach to AI in Aviation*; EASA: Cologne, Germany, 2023.
129. Leveson, N.G. *Engineering a Safer World: Systems Thinking Applied to Safety*; The MIT Press: Cambridge, MA, USA, 2012; ISBN 978-0-262-29824-7.
130. Sulaman, S.M.; Beer, A.; Felderer, M.; Höst, M. Comparison of the FMEA and STPA Safety Analysis Methods–a Case Study. *Softw. Qual. J.* **2019**, *27*, 349–387. [CrossRef]
131. Ahlbrecht, A.; Durak, U. Model-Based STPA: Enabling Safety Analysis Coverage Assessment with Formalization. In Proceedings of the 2022 IEEE/AIAA 41st Digital Avionics Systems Conference (DASC), Portsmouth, VA, USA, 18–22 September 2022; pp. 1–10.
132. Thomas, J.P.; Van Houdt, J.G. *Evaluation of System-Theoretic Process Analysis (STPA) for Improving Aviation Safety*; Federal Aviation Administration: William J. Hughes Technical Center, Systems Safety Section: Atlantic City, NJ, USA, 2024. [CrossRef]
133. Cofer, D.; Amundson, I.; Sattigeri, R.; Passi, A.; Boggs, C.; Smith, E.; Gilham, L.; Byun, T.; Rayadurgam, S. Run-Time Assurance for Learning-Enabled Systems. In *Proceedings of the NASA Formal Methods*; Lee, R., Jha, S., Mavridou, A., Giannakopoulou, D., Eds.; Springer International Publishing: Cham, Switzerland, 2020; pp. 361–368.
134. Hobbs, K.L.; Mote, M.L.; Abate, M.C.L.; Coogan, S.D.; Feron, E.M. Runtime Assurance for Safety-Critical Systems: An Introduction to Safety Filtering Approaches for Complex Control Systems. *IEEE Control Syst. Mag.* **2023**, *43*, 28–65. [CrossRef]
135. Woods, D.D. The Theory of Graceful Extensibility: Basic Rules That Govern Adaptive Systems. *Environ. Syst. Decis.* **2018**, *38*, 433–457. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.