Quantifying AI Failures for Integration into Physical AI Reliability Models

Original Scientific Paper

Milan Z. Bjelica*
University of Novi Sad,
Faculty of Technical Sciences, Department for Computer Engineering and Computer Communications
Trg Dositeja Obradovica 6, Novi Sad, Serbia
milan.bjelica@uns.ac.rs

Bogdan Pavković
University of Novi Sad,
Faculty of Technical Sciences, Department for Computer Engineering and Computer Communications
Trg Dositeja Obradovica 6, Novi Sad, Serbia
bogdan.pavkovic@uns.ac.rs

Denis Vranješ
J. J. Strossmayer University of Osijek,
Faculty of Electrical Engineering, Computer Science and Information Technology
Kneza Trpimira 2B, Osijek, Croatia
dvranjes@ferit.hr

*Corresponding author

Abstract *– Physical Artificial Intelligence (Physical AI) brings learning-enabled functions into systems that interact directly with the physical world. Examples range from autonomous robots and mobile platforms to industrial machinery and road vehicles. Reliability of the physical part of such systems is commonly addressed using established methods based on failure rates, reliability models, diagnostics, redundancy, and common-cause analysis. The behavior of AI components is treated differently. AI-related failures are usually evaluated through testing, scenario coverage, verification, operational monitoring, and safety argumentation, which makes their direct integration into classical system reliability models difficult. This paper proposes a quantitative representation of AI behavioral failures that can be used together with conventional reliability engineering. An AI failure is defined as an observed violation of an allocated automation task under specified operating conditions. The occurrence of these violations is represented by a counting process. Under a non-homogeneous Poisson model, the corresponding failure frequency is shown to be equal to the failure rate of an equivalent AI reliability block. This makes it possible to express AI behavioral performance in failures per hour and, where useful, in equivalent PFH or FIT units. AI components can consequently be included in reliability block diagrams together with hardware, mechanical, and conventional control-system components. AI updates are represented as behavioral renewal epochs, while a β_{AI} factor is introduced to account for the continuous monitoring and inclusion of edge cases appearing in the long tail of distribution with AI verification. Analytical derivation and representative examples illustrate the applicability and limitations of the proposed approach.*

Keywords: *Physical AI, AI safety, reliability, functional safety, failure intensity, failure frequency, reinforcement learning, renewal process, reliability block diagram.*

Received: August 16, 2026; Received in revised form: September 21, 2026; Accepted: September 21, 2026

1. INTRODUCTION

Artificial intelligence is increasingly becoming part of systems that do more than process information. It is being integrated into machines that perceive their surroundings, make decisions, and act in the physical environment. Industrial robots, service robots, autonomous mobile robots, automated machinery, drones, and road vehicles are representative examples. This wider class of systems is increasingly described as Physical AI.

An unsuitable AI decision in a Physical AI system can immediately result in physical motion or another action with consequences for people, equipment, infrastruc-

ture, or the environment, therefore, safety challenges with it are immense. At the same time, the system still contains conventional components familiar from reliability and safety engineering: sensors, computing hardware, power electronics, mechanical elements, actuators, communication interfaces, and conventional control software.

This produces an interesting mismatch in the way reliability is treated.

For hardware and mechanical components, engineering has a well-established quantitative language. Failure rates, dangerous failures, diagnostic coverage, common-cause failures, series and parallel reliability structures, fault trees, and reliability block diagrams are routinely used. Functional-safety standards build on this foundation. IEC 61508 [1] provides the generic framework, IEC 62061 [2] and ISO 13849 [3] address machinery, while ISO 26262 [4] provides a domain-specific implementation for road vehicles.

AI cannot be easily treated in similarly quantitative way. An AI component normally does not fail because its software physically wears out. Instead, it can remain fully operational from a computational point of view and still produce an inappropriate result. The model may incorrectly classify an object, fail to generalize to an unfamiliar condition, select an unsafe trajectory, or otherwise violate the task that was assigned to it. Standards and methods dealing with such problems therefore concentrate on scenarios, functional insufficiencies, data, robustness, verification, validation, and overall safety argumentation following the quality and safety processes.

Modelling physical components and AI components within a single framework would be of great use for continuous monitoring and tracking of the safety argument. Consider a robot in which an AI planner controls a physical platform. A reliability engineer may have failure-rate data for the electronic controller, sensors, motors, brakes, and power supply. The residual behavioral contribution of the AI planner, however, normally appears in a different form: test results, failed scenarios, simulation statistics, interventions, or other operational observations. A more straightforward quantification is needed that allows the AI-related evidence to be placed beside the physical component failure rates in the same reliability model, especially for AI systems that continue to evolve. A deployed AI-based system needs to be retrained, improved using data gathered during operation and updated over the air. Reinforcement-learning mechanisms can also change the system behavior between releases. The relevant safety argument is not static over the product lifetime and needs to be always visible and monitored.

Exhaustive enumeration of all possible future AI failures is not a scalable solution to this problem. The number of combinations of environment, system state, interaction, and edge conditions can become extremely large. Even a strong scenario-based argument cannot prove that every relevant situation has been identified.

This paper proposes a complementary behavioral view, acknowledging that the underlying deficiency of an AI component can remain systematic. However, the manifestation of that deficiency during operation depends on the conditions encountered by the system. When the system is observed over a sufficiently defined operational profile, these manifestations form an observable event process which can be represented by a quantity compatible with classical reliability engineering.

The proposed framework is built around four elements. First, an AI failure is defined at the level of the assigned automation task rather than at the level of an internal software defect. Second, failure frequency from the observed behavioral event process is related to the failure rate of an equivalent reliability block. Third, the resulting equivalent AI failure intensity can be combined with conventional component failure rates in reliability block diagrams. Finally, the approach explicitly distinguishes residual statistically treated behavior from the long tail of edge cases through a proposed β_{AI} factor. Methodological novelty of this approach lies in the explicit guidelines on how to reference, include and assess the AI failures in the context of the more classical quantitative safety frameworks, what is shown to be necessary for modern AI safety approaches such as with ISO/PAS 8800 [5].

Finally proposed framework is a contribution to the established AI safety engineering. Scenario analysis, validation, formal methods, monitoring, and structured safety argumentation remain necessary, whereas an additional quantitative interface between these activities and the reliability engineering already used for the physical part of Physical AI systems is established.

2. SAFETY, RELIABILITY AND RELATED WORK FOR PHYSICAL AI

2.1. RISK ACCEPTANCE

A failure rate is always regarded in the context of consequences of the corresponding failures which constitute hazards. Functional safety therefore starts from risk and from the level of risk that is considered tolerable.

IEC 61508 follows this principle by deriving the required safety integrity from the required reduction of risk. It also makes an important distinction between random hardware failures and systematic failures. Software-related deficiencies generally belong to the latter category.

Highly automated systems introduce applications in which automation replaces a task that was previously carried out by a human or by a less automated technical system. This creates both absolute and comparative ways of discussing acceptable risk.

The GAME/GAMAB principle, *Globalement Au Moins Aussi Bon*, expresses a relative criterion: a new system

should be at least as safe as an accepted reference system. This type of argument is particularly intuitive for automation replacing an existing human-operated solution.

The Minimum Endogenous Mortality (MEM) principle places an absolute limit on the contribution that a technical system may make to individual risk [6].

The two principles illustrate different questions. GAME asks whether automation is at least as good as what is already accepted. MEM asks whether the absolute contribution of the system remains sufficiently small.

These principles are useful for deciding whether the final system risk can be accepted, but become hard to decompose in practice to pinpoint the parts of the system which contribute most to the resulting risk. For a Physical AI architecture, these parts involve mechanics, electronics, sensing, conventional software, AI behavior, or a combination of these. A quantitative representation at this intermediate engineering level is therefore valuable even when the final risk acceptance is based on another criterion.

2.2. QUANTITATIVE RELIABILITY IN FUNCTIONAL SAFETY

Several safety domains already use quantitative reliability as part of safety-integrity demonstration.

IEC 61508 provides the generic basis for electrical, electronic, and programmable electronic safety-related systems. IEC 62061 specializes in this approach for machinery. ISO 13849-1 provides another widely used machinery framework based on Performance Levels and uses quantities such as dangerous failure frequency, $MTTF_D$, diagnostic coverage, architecture category, and common-cause considerations.

Robotic systems use this machinery-safety foundation in combination with more specific standards. ISO 10218 [7] addresses industrial robots, ISO 3691-4 [8] addresses driverless industrial trucks and automated guided or autonomous mobile platforms, while ISO 13482 [9] addresses personal-care robots.

Road vehicles provide another clear example through ISO 26262. Hardware safety evaluation uses probabilistic metrics that make it possible to reason about very small residual failure contributions. Typical PMHF target values associated with higher Automotive Safety Integrity Levels are of the order of 10^{-7} failures per hour for ASIL B and C and 10^{-8} failures per hour for ASIL D, corresponding to approximately 100 FIT and 10 FIT.

The exact numbers and metrics differ between industries, but the engineering principle is similar. Failure contributions are quantified, architecture is taken into account, and the resulting system-level performance is compared with an integrity requirement.

This quantitative representation enables architecture-level reasoning. Series contributions can be added, redundancy can be evaluated, diagnostic measures can receive appropriate credit, and common-cause failures can be separated from independent failures. There are many research approaches where such architectures are practically utilized [10].

2.3. AI SAFETY AND SOTIF

AI-related failures do not fit naturally into the random-hardware-failure interpretation and are mostly regarded separately before the system-level argumentation is built up. A neural network may execute correctly from a computational perspective but still produce an unsuitable decision. A perception system may miss an object because the situation lies outside the effective coverage of its learned representation. A planning function may select an unsafe maneuver because an unusual interaction was not sufficiently represented in training or validation.

In road vehicles, ISO 21448 [11] addresses this broader class of functional insufficiencies through Safety of the Intended Functionality. ISO/PAS 8800 [5] further develops a safety framework specifically for AI-based automotive systems. These approaches are necessary because conventional functional safety alone does not cover all causes of unsafe behavior in AI-enabled systems. They emphasize identification of insufficiencies, data quality, verification, validation, robustness, and supporting safety evidence. At the same time, they do not provide the metrics which would allow AI components to be modelled for failures on the architectural level.

This matters beyond automotive applications. A Physical AI robot may contain motors whose dangerous failure contribution can be estimated, sensors with established reliability data, electronic controllers with known hardware failure rates, and an AI function whose residual reliability is represented only through validation evidence. Practical robotic systems already illustrate this heterogeneous architecture. For example, Yousif and El-Medany [12] present an autonomous patrol robot that combines sensing, navigation, obstacle avoidance, communication, and physical mobility, demonstrating how intelligent functionality is integrated with conventional electronic and mechanical subsystems. From the system reliability perspective, one part of the architecture is consequently described in a different mathematical language.

The objective of this work is to provide a bridge between these two representations.

2.4. RELIABILITY OF LEARNING-ENABLED SYSTEMS

Several research directions already address quantitative reliability or failure assessment of learning-enabled systems.

Dong *et al.* [13] proposed a reliability assessment method for deep-reinforcement-learning-controlled robotic and autonomous systems. Their approach combines reachability verification with an operational pro-

file so that verified local regions contribute to an overall reliability estimate. This work is important because it shows that reliability concepts can be meaningfully applied to DRL-controlled Physical AI systems. However, the authors integrate reliability over verified regions of the state space, whereas in our work the goal is to derive an operational failure intensity from observed behavioral failures and make that quantity directly usable in a conventional time-domain reliability model.

Software reliability growth provides another relevant direction. Merkel applied established software reliability growth models to autonomous-vehicle disengagement data [14]. Musa-Okumoto and related models were investigated as descriptions of the decreasing rate of disengagement events during development. This suggests that the evolution of autonomous-system behavioral failures can, at least in some cases, be represented using reliability-growth theory.

The relationship between observed failures and learning is also increasingly explicit. DeFIX identifies failure scenarios of autonomous-driving policies and uses reinforcement learning to generate improved policies for those scenarios [15]. The sequence of operation, failure identification, learning, and subsequent deployment is particularly relevant to the renewal interpretation developed in this paper.

Recent testing research likewise questions whether general AI performance indicators are sufficient for safety evaluation. Failure-based testing of DRL agents, for example, focuses explicitly on task-induced failures rather than assuming that accumulated reward alone adequately identifies unsafe behavior [16].

Reinforcement learning is also increasingly used to provide adaptive decision-making outside autonomous vehicles and robotics. For example, the DRL-based approach in [17] dynamically adapts task scheduling in edge-cloud environments to changing system conditions. Although this work does not address functional safety or reliability directly, it illustrates a broader shift from fixed software behavior toward learned decision policies that adapt to complex operating conditions. For Physical AI, this trend further motivates reliability models that can represent behavioral performance rather than relying only on a fixed catalogue of software failure causes.

Although current research results address different elements of the same general problem, still there is little formal ground which ties up the AI behavior to the top level architectural view of the final system task.

2.5. RESEARCH GAP

Once there is evidence about the behavior of the AI components, collected through field testing or simulation, there is a clear research gap when this evidence needs to be brought to the reliability level.

This paper proposes the chain and the specific framework for the fulfillment of the following logical flow steps: (1) operational evidence; (2) behavioral failure process; (3) failure frequency; (4) equivalent AI failure intensity; (5) reliability composition; (6) system risk; (7) risk acceptance indication. In principle, the framework is not necessarily AI-specific and could also be applied to other automated processes. Therefore, the framework addresses physical AI as an application domain, suggesting the integration concepts and AI renewal linkage.

Note that the risk acceptance indication is not final. The additional requirements, that known systematic insufficiencies not disappear inside an aggregate statistical rate, still stand. However, continuous monitoring highly benefits fast trigger events which can be based on the early risk acceptance indication for the timely recall or update reaction.

3. BEHAVIORAL FAILURE MODEL FOR PHYSICAL AI

3.1. PHYSICAL AI SYSTEM MODEL

Consider a Physical AI system S composed conceptually of electronic hardware and sensing H, mechanical and electromechanical components M, conventional control and software C, and one or more AI-enabled components A:

$$S = \{H,M,C,A\} \tag{1}$$

This decomposition is formed from the standpoint of the type of reliability evidence that is available for the corresponding elements. With respect to architecture, AI and conventional software may run on the same processor, while individual physical components can combine mechanical and electronic elements.

For the conventional components, established component or subsystem reliability models can normally be used. The framework introducted in this paper derives an equivalent representation for A.

3.2. AUTOMATION-TASK VIOLATION

For an AI component A_i, consider the automation task allocated to that component. Depending on the system, this can include perception, navigation, object recognition, trajectory generation, manipulation, or another automated function.

In this paper, an AI behavioral failure is defined as an observed violation of an allocated automation task by an AI-enabled component or function under specified operating conditions.

This definition is behavioral and does not require knowledge of the internal software defect responsible for the observed outcome.

A safety-related subset can be defined as

$$F_{A,s} \subseteq F_A \tag{2}$$

where $F_{A,s}$ contains those automation-task violations that are relevant to a safety requirement.

In early automated-driving systems, disengagements offered a convenient observable indication that automated operation could not continue as intended. However, disengagement is tied to the presence of a safety driver and therefore cannot serve as a general Physical AI failure definition.

A task violation can instead be detected by an independent safety monitor, a task-level invariant, a runtime supervisor, post-processing of recorded data, a human intervention, or another predefined failure detection mechanism.

The failure definition is part of the model. If this definition changes, the corresponding rate cannot automatically be compared with a rate estimated using a different event definition.

3.3. OPERATIONAL EXPOSURE AND COUNTING PROCESS

Let $N(t)$ denote the number of qualifying automation-task violations observed during operational exposure t.

The mean cumulative number of failures is

$$W(t) = E[N(t)] \tag{3}$$

and failure frequency is defined as

$$w(t) = \frac{dW(t)}{dt} \tag{4}$$

when this derivative exists [18].

In our proposed framework, exposure variable is set to operating time because it permits direct comparison with failures-per-hour metrics. Other exposure measures, including operating cycles or distance traveled, can be used where they better describe the relevant task.

The evidence contributing to $N(t)$ can come from field operation, fleet data, structured testing, or simulation. These data sources should not automatically be treated as equivalent. Their contribution depends on how well they represent the operational profile for which the reliability claim is made.

4. EQUIVALENT FAILURE INTENSITY OF AI COMPONENTS

4.1. BEHAVIORAL-FAILURE POINT PROCESS

Consider one stable AI configuration over an analyzed interval. The relevant configuration includes the trained model and any other parameters that can materially change its observable behavior.

Assume that automation-task violations follow a non-homogeneous Poisson process with intensity

$$\nu(t) \geq 0 \tag{5}$$

and cumulative intensity

$$\Lambda(t) = \int_0^t \nu(u)\, du \tag{6}$$

For this process,

$$E[N(t)] = \Lambda(t) \tag{7}$$

and therefore

$$w(t) = \frac{dE[N(t)]}{dt} = \nu(t) \tag{8}$$

The Poisson assumption is important [20]. Although not every AI system necessarily exhibits this behavior, based on the previous research findings with disengagement events in automotive [14] as well with AI-based renewals [21] this model proves tractable allowing us to propose the reliability equivalence. This correlation would be more prominent in the upcoming end-to-end ML approaches, where automation task is tied more closely to the system level functional requirements, whereas with modular AI approaches a more elaborate decomposition might be required. In any case, the methodological framework here established always helps to assure early indication of the feasibility of the selected implementation approaches and architectures, where deep failure analysis is still unavailable.

4.2. AI RELIABILITY BLOCK

An equivalent AI reliability block can now be defined.

Success of this block over exposure t means that no qualifying automation-task violation occurs during that exposure. Its reliability is therefore

$$R_{AI}(t) = P(N(t) = 0) \tag{9}$$

For a non-homogeneous Poisson process,

$$R_{AI}(t) = e^{-\Lambda(t)} \tag{10}$$

or, using the cumulative intensity definition,

$$R_{AI}(t) = e^{-\int_0^t \nu(u)\, du} \tag{11}$$

The expression has the same form as the reliability function of a conventional non-repairable component with a time-dependent failure rate.

4.3. EQUIVALENCE BETWEEN FAILURE FREQUENCY AND EQUIVALENT FAILURE RATE

The failure rate of the equivalent AI block is defined as

$$\lambda_{AI,eq}(t) = -\frac{d}{dt} \ln R_{AI}(t) \tag{12}$$

Substituting (11),

$$\lambda_{AI,eq}(t) = -\frac{d}{dt}\left[-\int_0^t \nu(u)\, du\right] \tag{13}$$

which gives

$$\lambda_{AI,eq}(t) = \nu(t) \tag{14}$$

At the same time,

$$w(t) = \nu(t) \tag{15}$$

and consequently

$$\boxed{\lambda_{AI,eq}(t) = w(t)} \tag{16}$$

It should not be interpreted as a claim that failure frequency and hazard rate are universally identical quantities. For a general renewal process or arbitrary lifetime distribution this is not the case. The equality follows here from the chosen behavioral-event model and from defining reliability as the probability that no qualifying task violation occurs.

For the homogeneous special case,

$$\nu(t) = \nu_0 \tag{17}$$

the reliability becomes

$$R_{AI}(t) = e^{-\nu_0 t} \tag{18}$$

and

$$\lambda_{AI,eq} = w = \nu_0 \tag{19}$$

which is the conventional constant-rate exponential model.

4.4. PFH AND EQUIVALENT FIT REPRESENTATION

If exposure is measured in hours, the resulting equivalent rate has units

$$\lambda_{AI,eq}\ [h^{-1}] \tag{20}$$

and can be represented in equivalent FIT units as

$$FIT_{AI,eq} = 10^9 \lambda_{AI,eq} \tag{21}$$

For the safety-related subset of the behavioral failures,

$$\lambda_{AI,s,eq}(t) = w_s(t) \tag{22}$$

can analogously be used.

The word equivalent is deliberately retained. A value of $10^{-6}\,h^{-1}$ assigned to an AI block does not imply a random physical deterioration process equivalent to an electronic component with the same rate. It means that, under the declared event definition and operational profile, the AI behavioral process produces the same zero-failure reliability function as the corresponding reliability block.

4.5. ESTIMATION FROM OPERATIONAL EVIDENCE

For an interval in which a constant behavioral failure intensity is considered adequate, let n qualifying failures be observed during representative exposure T.

The estimated equivalent rate is

$$\hat{\lambda}_{AI,eq} = \hat{w} = \frac{n}{T} \tag{23}$$

and the corresponding FIT representation is

$$\widehat{FIT}_{AI,eq} = 10^9 \frac{n}{T} \tag{24}$$

This estimate is straightforward, but its uncertainty is important. A small number of observations can result in a wide confidence interval. In particular, observing zero failures does not establish a zero failure rate.

The approach should therefore not be interpreted as a way to demonstrate extremely small safety targets purely by accumulating test hours. Statistical evidence is one part of the safety argument and needs to be combined with verification, architectural measures, and other available evidence.

5. RELIABILITY COMPOSITION OF PHYSICAL AI

5.1. COMMON RELIABILITY REPRESENTATION

The main practical reason for introducing the equivalent AI rate is that it allows the AI function to participate in the same reliability model as the physical system [22].

Consider a simplified Physical AI function containing a mechanical subsystem M, electronic hardware H, and an AI component A. For independent series blocks,

$$R_{sys}(t) = R_M(t)\, R_H(t)\, R_{AI}(t) \tag{25}$$

If M and H are represented by constant failure rates and the AI rate is time dependent,

$$R_{sys}(t) = e^{-(\lambda_M+\lambda_H)t-\int_0^t \lambda_{AI,eq}(u)\,du} \tag{26}$$

For a constant AI rate during the considered interval,

$$\lambda_{sys} = \lambda_M+\lambda_H+\lambda_{AI,eq} \tag{27}$$

The benefit of (27) is mainly architectural. It allows the relative contribution of the AI function to be considered beside sensing, electronics, mechanics, or other conventional elements.

5.2. REDUNDANCY AND DIVERSE AI CHANNELS

Reliability engineering becomes particularly useful when architectural measures are introduced.

For two AI channels A_1 and A_2 performing the same function, full statistical independence would give

$$R_{A,parallel}(t) = 1-[1-R_{A1}(t)][1-R_{A2}(t)] \tag{28}$$

This can describe architectures containing diverse perception systems, an AI function and an independent AI monitor, or two independent implementations of the same automation function.

However, behavioral independence is much more difficult to justify than implementation diversity.

Two or more AI systems can use different implementations and still fail under the same situation because they share training data, sensors, environmental assumptions, task specifications, model families, or similar learned representations. Especially in the presence of unknown unknowns, as required in [10], it is possible that the reliability calculation breaks under new findings, where failures might accumulate and risk dramatically increase instantaneously.

5.3. THE β_{AI} FACTOR AND EDGE CASES

Classical redundant-system analysis has a similar problem when multiple components are vulnerable to a

common cause. The β-factor model provides one way of separating independent and common-cause contributions [19]. In the systems where AI is heavily utilized, edge cases lie at the edges of the behavioral failure observation distribution, potentially leading to high severity hazards. Therefore, they also need to be adequately modeled.

We use classical concept as an inspiration for a related concept which allows us to quickly update the reliability estimation when new edge cases are discovered. As shown in Fig. 1, failure rates of the AI components are adjusted and new edge cases are regarded separately as long as the update of the implementation is not performed and statistical ground truth established for the complete automation task as before the edge case discovery. Behavioral failures in the presence of edge cases are immediately extracted from the average AI failure rate approximation upon discovery, and treated separately as long as sufficient ground is established to prove that the behavioral failures under edge cases can be placed under the same (yet updated) overall distribution.

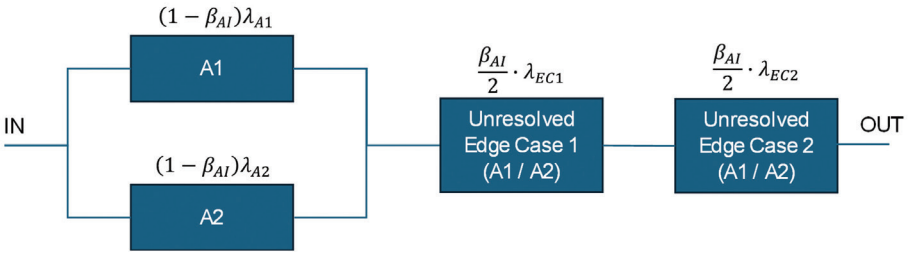


Fig. 1. Reliability block diagram for diverse configuration of AI components where β_{AI} is used to account for the appearance of newly discovered edge cases

For each AI channel with equivalent behavioral failure intensity λ_{An}, define

$$\lambda_{An,I} = (1-\beta_{AI})\lambda_{An} \tag{29}$$

And for each newly discovered edge case

$$\lambda_{EC,I} = \frac{\beta_{AI}}{K}\lambda_{EC} \tag{30}$$

Where K is the number of still unsolved edge cases, and also

$$0 \le \beta_{AI} \le 1 \tag{31}$$

Assuming both AI components have the same failure rates, the reliability of the independent parallel contribution is

$$R_{AI}(t) = 1-[1-e^{-(1-\beta_{AI})\lambda_A t}]^2 \tag{32}$$

while in case of a single new edge case found

$$R_{EC}(t) = e^{-\beta_{AI}\lambda_{EC}t} \tag{33}$$

The resulting reliability for this example is

$$\boxed{R_{\beta}(t) = e^{-\beta_{AI}\lambda_{EC}t}\{1-[1-e^{-(1-\beta_{AI})\lambda_A t}]^2\}} \tag{34}$$

The analogy to the classical common-cause β factor allows the classical calculus to be performed when AI components are considered. It is important, however, to be able to correctly estimate λ_{EC} based on the newly discovered data describing the edge case, and also to determine β_{AI} based on the proof of exposure to the edge case. In the presence of newly discovered edge case, both β_{AI} and λ_{EC} shall be set conservatively until additional ground truth proof is obtained. This allows the most accurate final reliability and safety assessment of the existing configuration and facilitates the decision on the system recall and the timeline (urgency) of the following update.

6. CONTINUOUS RELIABILITY ARGUMENTATION THROUGH AI RENEWAL

6.1. BEHAVIORAL RENEWAL EPOCHS

The AI-based functionality differs from many conventional components because its behavior can change materially during the system lifetime.

Retraining, reinforcement learning, calibration changes, software updates, and OTA deployment can produce a new behavioral configuration without replacing the physical hardware.

Let

$$T_0, T_1, \ldots, T_k, \ldots \tag{35}$$

denote the release times of behaviorally relevant AI versions.

One behavioral epoch can be defined as:

$$E_k = [T_k, T_{k+1}) \tag{36}$$

Within this interval, the corresponding AI configuration is assumed to be stable enough for its failure process to be characterized. When a new release materially changes the behavior, a new epoch starts.

This idea follows the renewal interpretation previously used for software upgrades and AI-based reconfiguration. In the earlier formulation, software upgrade or self-reconfiguration is treated as repair-like behavior in the context of software reliability growth, including changes triggered by reinforcement learning [23][24]. Periodic upgrades were also distinguished from instantaneous updates, with component behavior remaining unchanged between successive upgrades.

The refinement introduced here is to separate the two processes clearly. Automation-task violations form the

behavioral **counting process inside an epoch**, while AI releases define the **renewal process between epochs**.

If the inter-update interval is represented by

$$Y_k = T_{k+1} - T_k \tag{37}$$

then a classical renewal model can be used when the appropriate assumptions hold. In practical deployed AI systems, update timing will often depend on accumulated evidence and engineering decisions, so a generalized renewal interpretation can be more appropriate.

6.2. REINFORCEMENT LEARNING AND RELIABILITY GROWTH

The renewal view is particularly suitable for AI systems that improve using operational evidence. A simplified lifecycle can be represented as the sequence of steps: (1) operation; (2) violation detection; (3) data acquisition; (4) learning/retraining; (5) verification; (6) release and (7) operation. This cycle has practical analogues in continual-learning research. DeFIX [15], for example, identifies failure scenarios in autonomous driving and uses them to train additional reinforcement-learning policies.

The equivalent failure intensity then becomes version-dependent:

$$\lambda_{AI,0} \rightarrow \lambda_{AI,1} \rightarrow \cdots \rightarrow \lambda_{AI,k} \tag{38}$$

A successful improvement process would normally aim for

$$\lambda_{AI,k+1} < \lambda_{AI,k} \tag{39}$$

but this relation is not assumed by the framework. An update can introduce a regression, change the effective operational domain, or expose a previously irrelevant failure mode. The new rate therefore needs to be supported by new evidence.

This provides a simple way of connecting reliability growth with continuously evolving Physical AI.

6.3. FROM RELIABILITY TO RISK ACCEPTANCE

The complete quantitative chain can now be written as

$$\underbrace{\text{fleet/simulation evidence}}_{\rightarrow N_k(t) \rightarrow w_k(t) \rightarrow \lambda_{AI,k,eq}(t) \rightarrow RBD \rightarrow \lambda_{sys}(t) \rightarrow} \underbrace{\text{risk acceptance}} \tag{40}$$

The last step remains application dependent.

A GAME-type argument can compare the resulting system behavior with an accepted reference. A MEM-type argument can instead constrain the absolute contribution to risk.

The important point is that risk acceptance does not need to rely only on final accident statistics. Intermediate quantitative arguments become possible. If system reliability is dominated by the AI function, additional AI improvement or architectural diversity may be appropriate. If the AI contribution becomes small compared with actuator or hardware failures, further AI improvement may have limited influence on the final system reliability.

7. ANALYTICAL VERIFICATION WITH EXAMPLES

The analytical consistency of the proposed representation follows directly from the relations established in Sections 3 and 4. Under the assumed NHPP model, Eq. (16) establishes that the equivalent AI failure rate is equal to the behavioral failure frequency. The expected limiting behavior is also preserved. A constant behavioral failure frequency results in the conventional exponential reliability model, while a decreasing failure frequency results in a decreasing equivalent failure intensity. A behaviorally significant AI update terminates the current epoch and starts a new estimation period.

The expected limiting behavior also follows directly. A constant $w(t)$ gives an exponential reliability model. A decreasing $w(t)$ gives decreasing equivalent failure intensity. A behaviorally significant AI update terminates the current epoch and starts a new estimation period.

Let's now use an example to illustrate reliability evolution across AI releases. Consider a simplified Physical AI mobile platform. Assume that the combined failure intensity associated with the non-AI part of the system is

$$\lambda_{base} = 1.0 \times 10^{-6}\ h^{-1} \tag{41}$$

Five consecutive AI versions are considered, with decreasing illustrative behavioral failure intensities.

For the series reliability model,

$$\lambda_{sys,k} = \lambda_{base} + \lambda_{AI,k} \tag{42}$$

and

$$R_{sys,k}(t) = e^{-\lambda_{sys,k} t} \tag{43}$$

Table 1 gives the resulting values for a reference exposure of 1000 h.

Table 1. Illustrative reliability evolution across AI renewal epochs

Release	λ_{AI}[1/h]	AI equiv. FIT	λ_{sys}[1/h]	System equiv. FIT	R_{sys}(1000h)
A0	5.0×10^{-5}	50,000	5.1×10^{-5}	51,000	0.950279
A1	2.0×10^{-5}	20,000	2.1×10^{-5}	21,000	0.979219
A2	8.0×10^{-6}	8,000	9.0×10^{-6}	9,000	0.991040
A3	3.0×10^{-6}	3,000	4.0×10^{-6}	4,000	0.996008
A4	1.0×10^{-6}	1,000	2.0×10^{-6}	2,000	0.998002

The example shows that at the beginning, system reliability is dominated by the AI behavioral failure contribution. As the AI component improves, the relative contribution of the conventional part becomes increasingly important. In release A4, the assumed AI rate has reached the same order as the entire non-AI contribution.

This type of calculation can therefore support engineering prioritization. It indicates whether improving

AI behavior is still the most effective way of improving the final system or whether another part of the architecture has become the limiting contributor.

The second example considers two diverse AI channels with their respective failure rates and reference mission T:

$$\lambda_{AI} = 1.0\times10^{-5}\ h^{-1} \tag{44}$$

$$T = 1000\ h \tag{45}$$

Using (34), the redundant system reliability can be evaluated for different assumptions about β_{AI}.

Because the resulting redundant structure does not generally have a constant exponential rate, a mission-equivalent rate and corresponding FIT can be defined as

$$\lambda_{\beta,eq}(T) = -\frac{lnR_{\beta}(T)}{T} \tag{46}$$

$$FIT_{\beta,eq}(T) = 10^{9}\lambda_{\beta,eq}(T) \tag{47}$$

Representative results are given in Table 2.

Table 2. Effect of β_{AI} on dual-channel AI reliability

β_{AI}	R_{β} (1000h)	Failure probability	Equivalent rate [1/h]	Equivalent FIT
0.00	0.999901	0.000099	9.90×10^{-8}	99
0.05	0.999411	0.000589	5.89×10^{-7}	589
0.10	0.998920	0.001080	1.08×10^{-6}	1,080
0.20	0.997939	0.002061	2.06×10^{-6}	2,063
0.50	0.994988	0.005012	5.02×10^{-6}	5,025
1.00	0.990050	0.009950	1.00×10^{-5}	10,000

At $\beta_{AI} = 0$, complete independence is assumed and two AI channels provide a large reliability improvement. Once an edge case is discovered, the mission-equivalent rate may increase depending on the edge case exposure estimation for each selected β_{AI}, significantly reducing the output reliability and increasing the risk as long as edge cases are unresolved.

Although full empirical evaluation for the final applicability to the failure outcomes in a real system is still needed to fully confirm the proposed approach, the apparatus given in the paper allows a reasonable upper-bound estimation as well as the methodology for edge case treatment which can be used as argumentation within a safety process in the scope of ISO/PAS 8800, where these aspects are left largely underspecified.

8. DISCUSSION AND LIMITATIONS

8.1. SYSTEMATIC FAILURES VS AI "RANDOMNESS"

The most obvious question challenging the proposed approach is that AI is implemented in software, while software failures are systematic. This criticism is valid if the equivalent failure intensity is interpreted as the failure rate of the underlying software defect. Although software deficiency can be perfectly deterministic and given the same inputs and state, it may reproduce exactly the same failure, what varies during operation is the exposure of that deficiency to environmental and system conditions.

The stochastic element modeled in this paper is therefore the occurrence of the behavioral manifestation, what is an important distinction. It allows operational behavior to be treated statistically without contradicting the systematic-failure classification used in functional safety.

8.2. ASSUMPTION ON THE FAILURE PROCESS

The equality in (16) is derived under an NHPP assumption, what is used as a tractable model for reliability estimation, establishing a reactive framework to ever-changing ground truth in AI verification. In general, this model cannot be assumed for every AI system. Behavioral failures may cluster. They may depend on operating mode or environmental condition. One failure may influence the likelihood of another. Different operational regimes may also exhibit very different failure intensities. In such cases, a more suitable point-process model should be used.

The broader concept where an equivalent AI reliability block can be defined using the probability of zero task violations remains applicable. However, the direct equality between failure frequency and equivalent hazard must then be re-established for the selected statistical model.

8.3. DEPENDENCE ON THE OPERATIONAL PROFILE

An equivalent AI rate cannot be separated from the operational conditions under which it was measured.

A navigation model operating in an empty warehouse does not necessarily have the same behavioral failure intensity when the same warehouse contains people, moving forklifts, temporary obstacles, or poor illumination. Every reliability claim should consequently identify the relevant automation task, AI version, operational domain, event definition, and source of exposure evidence. This also limits how simulation evidence can be used. Large, simulated exposure is useful only to the extent that the simulated conditions represent the behavior relevant to the safety claim.

8.4. VERY LOW FAILURE RATES

Safety-integrity targets can involve failure frequencies that are too small to establish directly through realistic operational testing. The proposed equivalent failure-rate formulation does not remove this difficulty. For example, observing no qualifying failure in a large but finite number of hours produces an upper statistical confidence bound; it does not prove that the true rate is zero or that an extremely small safety target has been reached. This is why the proposed measure

should remain part of a larger safety argument. Formal verification, scenario-based evidence, structural safety mechanisms, monitors, diagnostic coverage, safety envelopes, and process assurance remain important.

8.5. SELECTION OF THE β_{AI} FACTOR

β_{AI} is conceptually related to common-cause reliability modeling but estimating it will be more difficult than applying an established hardware beta-factor model.

AI channels can share several sources of dependence: training data, sensor inputs, requirements, architectures, reward functions, feature representations, model families, or environment assumptions. Even nominally diverse implementations can therefore exhibit significant co-failure behavior, but not necessarily completely the same when the new edge cases are discovered, so these aspects require additional research. At this moment, the parameter should be treated conservatively and be used as a vessel to prevent unjustified redundancy credit and allow visibility of immediate new risk and facilitate the recall decision.

8.6. INTEGRATION WITH SAFETY CASE

The statistical and systematic parts of AI safety should not be separated completely. An unidentified task violation contributes evidence to the residual behavioral process. Once the underlying situation has been identified as a repeatable safety-relevant insufficiency, its treatment should change. It becomes a known issue that requires explicit mitigation or limitation. Therefore, combination of functional safety approaches (AI "caging") remains together with the statistical data-driven AI performance indicators which may trigger automation task violation and induce failures.

The proposed framework is also intended to serve both general reliability estimation with respect to system functional requirements, as well as for safety requirements where the same metrics are used for safety-related failure paths. In this sense, safety-related behavioral failure needs to be regarded as one leading to the dangerous failure, what allows the reduction of failure rates in contrast with the complete automation task. This is common in safety engineering, where hardware failure rates in reliability block diagrams which are used account only for dangerous undetectable failures, in cases where those diagrams explain safety related systems.

9. CONCLUSION

Physical AI combines conventional physical components with learning-enabled functions, but these two parts are currently difficult to treat within one quantitative reliability model. The paper proposed a behavioral approach in which an AI failure is defined as a violation of the allocated automation task under specified operating conditions. The occurrence of such failures is represented by a counting process. Under an NHPP model, the failure frequency of that process was shown to equal the failure rate of an equivalent AI reliability block, what is useful as an estimation of an AI behavior quantity that can be expressed in failures per hour and, where appropriate, as an equivalent PFH or FIT value.

This allows AI components can be combined quantitatively with mechanical, electronic, and other conventional control components using established reliability techniques, allowing the preliminary framework to start building up the required proof for, i.e. safety case in the scope of ISO/PAS 8800.

The approach was further extended to evolving AI. Model releases and OTA updates define new behavioral epochs, allowing the corresponding failure intensity to change as the system learns or is improved. Discovered edge cases are treated by the introduction of β_{AI} to adapt the calculus when it starts to drift with respect to the observed ground truth.

The proposed method therefore is a reactive framework allowing facilitated insight into overall system reliability and safety when AI components are used, and where the fast reaction to incidents is of paramount importance. It makes it possible to retain the established principles of AI safety while introducing a quantitative interface to classical reliability engineering.

Further work should address alternative behavioral point-process models, confidence bounds for low event frequencies, treatment of mixed simulation and field exposure, quantitative estimation of β_{AI}, and evaluation of the method using long-term operational data from different classes of Physical AI systems.

ACKNOWLEDGMENT

This research was funded by the Interreg IPA Croatia-Serbia program, under the project HR-RS00269 "Boosting Research and Innovation Capacities in Cybersecurity-oriented and Functional Safety-oriented Smart Mobility". It has also been supported in part by the Ministry of Science, Technological Development and Innovation of the Republic of Serbia (Contract No. 451-03-34/2026-03/200156) and the Faculty of Technical Sciences, University of Novi Sad, Serbia through project "Scientific and Artistic Research Work of Researchers in Teaching and Associate Positions at the Faculty of Technical Sciences, University of Novi Sad 2026" (No. 01-3609/1).

DECLARATION OF THE USE OF AI TOOLS

The authors used ChatGPT 5.6 Sol for language editing and formatting. All AI-assisted content was reviewed and edited by the authors, who take full responsibility for the integrity and accuracy of this work. No AI tool is listed as an author, and AI tools are not cited as primary sources.

REFERENCES

[1] International Electrotechnical Commission, "Functional Safety of Electrical/Electronic/Programmable Electronic Safety-Related Systems – Part 1: General Requirements", IEC 61508-1:2010, Switzerland, 2010.

[2] International Electrotechnical Commission, "Safety of Machinery – Functional Safety of Safety-Related Control Systems", IEC 62061, Switzerland, 2026.

[3] International Organization for Standardization, "Safety of Machinery – Safety-Related Parts of Control Systems – Part 1: General Principles for Design", ISO 13849-1:2023, Switzerland, 2023.

[4] International Organization for Standardization, "Road Vehicles – Functional Safety – Part 5: Product Development at the Hardware Level", ISO 26262-5:2018, Switzerland, 2018.

[5] International Organization for Standardization, "Road Vehicles – Safety and Artificial Intelligence", ISO/PAS 8800:2024, Switzerland, 2024.

[6] J. Braband, B. Evers, M. Kinas, L. Lindner, D. Mihailescu-Stoica, F. Rexin, F. Adebahr, B. Milius, H. Schäbe, "Risikoakzeptanzkriterien für das automatisierte Fahren auf der Schiene", Deutsches Zentrum für Schienenverkehrsforschung, Report 40/2023, 2023. doi: 10.48755/dzsf.230008.01 (in German)

[7] International Organization for Standardization, "Robotics – Safety Requirements – Part 1: Industrial Robots", ISO 10218-1:2025, Geneva, Switzerland, 2025.

[8] International Organization for Standardization, "Industrial Trucks – Safety Requirements and Verification – Part 4: Driverless Industrial Trucks and Their Systems", ISO 3691-4:2023, Geneva, Switzerland, 2023.

[9] International Organization for Standardization, "Robots and Robotic Devices – Safety Requirements for Personal Care Robots", ISO 13482:2014, Switzerland, 2014.

[10] E. Bouchetob, B. Nadji, "Boosting Reliability: A Comparative Study of Silicon Carbide (SiC) and Silicon (Si) in Boost Converter Design Using MIL-HDBK-217 Standards", International Journal of Electrical and Computer Engineering Systems, Vol. 15, No. 4, 2024, pp. 313-320. doi: 10.32985/ijeces.15.4.2

[11] International Organization for Standardization, "Road Vehicles – Safety of the Intended Functionality", ISO 21448:2022, Switzerland, 2022.

[12] T. Yousif, W. El-Medany, "Development and Hardware Implementation of IoT-Based Patrol Robot for Remote Gas Leak Inspection", International Journal of Electrical and Computer Engineering Systems, Vol. 13, No. 4, 2022, pp. 279-292. doi: 10.32985/ijeces.13.4.4

[13] Y. Dong, X. Zhao, S. Wang, X. Huang, "Reachability Verification-Based Reliability Assessment for Deep Reinforcement Learning Controlled Robotics and Autonomous Systems", IEEE Robotics and Automation Letters, Vol. 9, No. 4, 2024, pp. 3299-3306. doi: 10.1109/LRA.2024.3364471

[14] R. G. Merkel, "Software Reliability Growth Models Predict Autonomous Vehicle Disengagement Events", arXiv.1812.08901, 2018.

[15] R. Dagdanov, F. Eksen, H. Durmus, F. Yurdakul, N. K. Ure, "DeFIX: Detecting and Fixing Failure Scenarios with Reinforcement Learning in Imitation Learning Based Autonomous Driving", Proceedings of the 25th IEEE International Conference on Intelligent Transportation Systems, Macau, China, 8-12 October 2022, pp. 4215-4220. doi: 10.1109/ITSC55140.2022.9922209

[16] W. Lin, J. Meng, Z. Zheng, "Failure-Based Testing for Deep Reinforcement Learning Agents", Proceedings of the ACM on Software Engineering, Vol. 3, 2026, pp. 4048-4069. doi: 10.1145/3808185

[17] D. M. Rani, S. K. P, B. B. Jayasingh, "Deep Reinforcement Learning for Dynamic Task Scheduling in Edge-Cloud Environments", International Journal of Electrical and Computer Engineering Systems, Vol. 15, No. 10, 2024, pp. 837-850. doi: 10.32985/ijeces.15.10.3

[18] International Electrotechnical Commission, "International Electrotechnical Vocabulary (IEV) – Part 192: Dependability", IEC 60050-192:2015, entry 192-05-08, "Instantaneous Failure Intensity", Switzerland, 2015.

[19] International Electrotechnical Commission, "Functional Safety of Electrical/Electronic/Programmable Electronic Safety-Related Systems – Part 6: Guidelines on the Application of IEC 61508-2 and IEC 61508-3", IEC 61508-6:2010, Geneva, Switzerland, 2010.

[20] S. M. Ross, "Stochastic Processes", 2nd Edition, Wiley, 1996.

[21] J. Min, Y. Hong, C. B. King, W. Q. Meeker, "Reliability Analysis of Artificial Intelligence Systems Using Recurrent Events Data from Autonomous Vehicles", Journal of the Royal Statistical Society Series C: Applied Statistics, Vol. 71, No. 4, 2022, pp. 987-1013. doi: 10.1111/rssc.12564

[22] M. Rausand, A. Høyland, "System Reliability Theory: Models, Statistical Methods, and Applications", 2nd Edition, Wiley-Interscience, 2004. doi: 10.1002/9780470316900

[23] M. Z. Bjelica, B. Mrazovac, "Reliability of Self-Driving Cars: When Can We Remove the Safety Driver?", IEEE Intelligent Transportation Systems Magazine, Vol. 15, No. 4, 2023, pp. 46-54. doi: 10.1109/MITS.2023.3244271

[24] M. Z. Bjelica, "Reliability Estimation for a Central Vehicle Computer with Upgradeable Software Components", University of Applied Sciences – FH Campus Wien, Vienna, Austria, Master Thesis, 2021.