INTL JOURNAL OF ELECTRONICS AND TELECOMMUNICATIONS, 2026, VOL. 72, NO. 3, PP. 1-9
Manuscript received March 17,2026; revised June, 2026. doi: 10.24425/ijet.2026.157952

Extended multi-fragment Markov models of programmable devices with controlled degradation considering hidden failures

Vyacheslav Kharchenko, Yuriy Ponochovnyi, Oleksandr Vdovichenko,
Valeriy Dubnitskiy, and Viktoriia Medvid

***Abstract*—This paper presents an enhanced Markov modelling framework for evaluating the dependability and instantaneous availability of programmable devices (PDs) operating with controlled multi-level degradation in aggressive environments. Two advanced multi-fragment models are introduced: MFM03, which generalises the topology to arbitrary trapezoidal/rectangular fragment arrangements using a uniform rectangular implementation, and MFM04, which additionally accounts for imperfect diagnostic coverage and hidden failures. The study describes the unified classification of the complete model family (SFM, MFM01, MFM02, MFM03, MFM04), state-space generation, transition matrix construction, and fully parametric MATLAB simulation approach. Numerical results based on ATtiny13A failure rates show that the generalised topology of MFM03 yields a 1–1.5 % absolute long-term availability improvement over the original strict triangular MFM02 after 3×10^6 hours. The MFM04 model, under realistic diagnostic coverage of 60–90 %, exhibits a crossover phenomenon whereby moderate diagnostic imperfection imposes a small medium-term penalty but can produce marginal ultra-long-term gain due to temporally distributed recovery actions. Overall, the extended model hierarchy achieves cumulative availability gains of 11–12 % compared to conventional binary single-fragment approaches, delivering practical, scalable tools for resilience optimisation of PDs in ultra-long-lifetime safety- and mission-critical systems including spacecraft onboard electronics, UAV control platforms, and so on.**

***Keywords*—Dependability; Availability; Programmable Devices; Markov Models; Multi-Level Degradation; Reconfiguration; Trapezoidal/Rectangular Topology; Hidden Failures; Imperfect Diagnostic Coverage; MATLAB Simulation**

I. Introduction

THE rapid proliferation of programmable devices (PDs) such as field-programmable gate arrays (FPGAs), microcontrollers, and system-on-chip solutions in safety- and mission-critical applications has significantly increased the demands placed on their dependability under extreme operational conditions. Systems operating in harsh environments – including unmanned aerial vehicle (UAV) fleets in contested or cyber-threatened areas [1,2], spacecraft exposed to high radiation, nuclear power plant instrumentation and control (I&C) platforms, demining robots in contaminated zones, and deep-sea or arctic sensor nodes – must maintain acceptable levels of functionality for years or decades despite progressive degradation of hardware resources caused by radiation-induced single-event effects, thermal stress, electromagnetic interference, vibration, aging, or intentional cyber-physical attacks [3,4]. Traditional binary reliability models that assume only two states (fully operational or completely failed) are inadequate for such applications because real programmable devices can continue to deliver reduced but still useful functionality after cumulative deterioration of technical characteristics with possible reduction in the number of performed functions – a properly known as multi-level degradation (MLD) [5].

To address this challenge, a reconfiguration approach to bypass the consequences of errors and failures emergence by software redistribution of the device's resources was considered. A combination of on-the-fly reconfiguration and controlled multi-level degradation has emerged as a promising approach for extending mission time in resource-constrained critical systems [6–8]. By dynamically reallocating tasks among remaining functional fragments of an FPGA or microcontroller, the system can gracefully degrade instead of suffering abrupt failure, thereby increasing overall availability in environments where repair is impossible or extremely delayed.

In our previous work [9], we observed approaches for assessing the dependability and instantaneous availability of programmable devices (PDs). According to existing issues of provided methods we proposed a family of Markov models for PD with controlled MLD. Three baseline models were developed and analysed using MATLAB simulations: (i) a single-fragment model (SFM) without reconfiguration capability; (ii) a multi-fragment model with degradation only (MFM01); and (iii) a multi-fragment model incorporating both degradation and reconfiguration (MFM02). Those models demonstrated that introducing controlled degradation and reconfiguration can substantially improve long-term availability compared to conventional single-fragment approaches, especially for operational lifetimes exceeding several decades.

However, the MATLAB implementation of MFM02 and the underlying conceptual topology impose certain restrictions that limit its applicability to real asymmetric degradation scenarios. Specifically, the realisation of MFM02 creates a strictly triangular fragment arrangement, leading to reduced modelling flexibility when degradation rates of programmable resources

Vyacheslav Kharchenko and Oleksandr Vdovichenko are with the National Aerospace University "Kharkiv Aviation Institute", Ukraine (e-mail: v.kharchenko@csn.khai.edu, o.vdovichenko@csn.khai.edu).

Valeriy Dubnitskiy is with the Karazin Banking Institute of V. N. Karazin Kharkiv National University, Ukraine (e-mail: dubnitskiy@gmail.com).

Yuriy Ponochovnyi and Viktoriia Medvid are with the Poltava State Agrarian University, Ukraine (e-mail: yuriy.ponch@gmail.com, viktoriia.medvid@pdau.edu.ua).

and reconfiguration circuitry differ arbitrarily. Moreover, all baseline models assume ideal fault detection and diagnostic coverage. In practice, fault coverage is imperfect, especially for single-event effects and latent failures in redundant reconfiguration paths. Hidden (undetected) failures can accumulate silently, leading to sudden loss of controllability – a scenario particularly dangerous in safety-critical applications [10,11]. Recent studies of IoT-enabled UAV communication services [12] and smart surveillance systems [13] further confirm that imperfect diagnostics and hidden cyber-physical faults significantly degrade long-term dependability in distributed programmable systems.

Additional motivation for model extension arises from recent field data and standards. Analyses of FPGA-based satellite systems [14] and UAV swarms [1] show that the number of usable degradation levels frequently exceeds three and is often asymmetric, requiring more general topologies than the strict triangular structure of MFM02. Moreover, IEC 61508 [15] and DO-254 [16] explicitly require quantitative assessment of diagnostic coverage and latent failure contributions when claiming high safety integrity levels (SIL 3–4 or DAL A–B). Existing multi-state system (MSS) theory [17,18] and universal generating function (UGF) techniques [19] provide valuable foundations but do not offer ready-to-use Markov constructs that simultaneously support arbitrary fragment topologies, imperfect fault detection, and emerging cyber-physical threats in modern UAV fleets and CubeSats [2,12,13,20].

Goal of the study is to develop the models providing accurate and trustworthy assessing reliability and availability of PDs with MLD considering hidden failures of components and means of reconfiguration. Therefore, the present paper significantly extends the framework introduced in [9] by developing two advanced multi-fragment Markov models:

– MFM03 – a generalised version of MFM02 employing a uniform rectangular n×m fragment grid with zero-intensity transitions to exclude inactive fragments, enabling arbitrary trapezoidal/rectangular topologies while keeping exactly five states per fragment;

– MFM04 – a further refinement that explicitly incorporates hidden degradation and hidden critical failures via additional states S_{hd} and S_{hf}, parameterized by diagnostic coverage factors D_d and D_f.

The generalised topology of MFM03 allows modelling systems with extended resource reserves, including battery-constrained CubeSats [20]. The paper is organized as follows. Section 2 recalls the assumptions and state notation from [9] and introduces new attributes for generalised topologies and hidden failures. Section 3 describes the MFM03 model and its MATLAB implementation. Section 4 presents MFM04 with imperfect diagnostics. Section 5 compares all five models and discusses implications for UAV control, space electronics, and nuclear I&C. Section 6 concludes and outlines future work.

II. Classification and General Assumptions

This paper extends the classification framework presented in [9] while preserving the fundamental distinction between single-fragment models (SFM) and multi-fragment models (MFM).

To support more flexible modelling of asymmetric and trapezoidal degradation scenarios, two additional classification criteria are introduced for the new models MFM03 and MFM04. Table I summarises the characteristics of all five models developed to date.

TABLE I
COMPARATIVE CLASSIFICATION OF MARKOV AVAILABILITY MODELS FOR PROGRAMMABLE DEVICES WITH CONTROLLED MULTI-LEVEL DEGRADATION

Attribute	SFM	MFM01	MFM02	MFM03 (enhanced MFM02)	MFM04 (new)
Number of fragments	1	Variable (1D)	Variable (2D)	Variable (2D)	Variable (2D)
Conceptual fragment topology	–	Linear	Strictly triangular	Arbitrary trapezoidal/rectangular	Arbitrary trapezoidal/rectangular
MATLAB implementation topology	–	Linear	Rectangular (with zero-weighted inactive parts)	Rectangular (uniform, zero-weighted inactive parts)	Rectangular (uniform, zero-weighted inactive parts)
Reachable fragments	1	Up to n	Up to ~n×(n+1)/2	Up to n×m (user-defined)	Up to n×m (user-defined)
Uniform state structure per fragment	Yes	Yes	No (boundary fragments have 4 states)	Yes (5 states in every fragment)	Yes (7 states in every fragment)
Reconfiguration capability	No	No	Yes	Yes	Yes
Diagnostic coverage	Perfect	Perfect	Perfect	Perfect	Imperfect (Dd, Df)
Hidden failure states (Shd, Shf)	No	No	No	No	Yes
Typical total states	4	12	40	60	84
MATLAB builder function	fM0.m	fM01.m	fM02.m	fM03.m	fM04.m

The key advancement in MFM03 compared to MFM02 lies in the MATLAB implementation and resulting modelling flexibility:

– MFM02 conceptually enforces a strict triangular arrangement and uses varying state counts per fragment (4 or 5), which complicates automated matrix generation.

– MFM03 adopts a uniform rectangular n×m grid in MATLAB where every fragment contains exactly five states (Su, Sd, Se, Sf, Sr). Inactive fragments and unnecessary transitions/states are excluded by assigning zero intensity to the corresponding entries in the λ_d and λ_r matrices. This unified structure dramatically simplifies code maintenance, numerical stability, and extension to arbitrary trapezoidal topologies (e.g., several rows with identical length followed by progressively shorter rows). Reconfiguration transitions move the system downward to a lower row in the fragment grid, consuming

reconfiguration reserve and — in trapezoidal topologies — reducing the number of available columns in subsequent fragments, thereby restricting further horizontal degradation transitions (zero intensity to inactive fragments) and limiting possible subsequent degradation states.

– MFM04 inherits the same uniform rectangular foundation from MFM03 and adds two hidden states (S_{hd}, S_{hf}) to every fragment, controlled by diagnostic coverage parameters D_d and D_f.

All models continue to use the baseline failure rates derived from the ATtiny13A microcontroller [9]. The unified set of input parameters employed in the current study is presented in Table II (values in $\times 10^{-5}$ h^{-1} unless stated otherwise).

TABLE II
INPUT PARAMETERS USED FOR DEPENDABILITY ASSESSMENT MODELS OF PROGRAMMABLE DEVICES

No	Parameter	Symbol	Value ($\times 10^{-5}$ h^{-1})	Used in models
1	Elementary pin failure rate	λpin	0.03	All
2	Core (tiny) failure rate	λtiny	0.002	All
3	Intensity of first degradation level	λdx0	4 × λpin = 0.12	All
4	Intensity of second degradation level	λdx1	3 × λpin = 0.09	MFM01–MFM04
5	Intensity of third degradation level	λdx2	2 × λpin = 0.06	MFM01–MFM04
6	Critical element failure intensity	λf	4 × λpin + λtiny = 0.122	All
7	Error (transient fault) intensity	λe	8 × λpin + λtiny = 0.242	All
8	Recovery intensity after errors	μe	1 666.67 (≈1/60 h^{-1})	All
9	Recovery intensity after reconfiguration/degradation	μu = μr	1 666.67 (≈1/60 h^{-1})	MFM02–MFM04
10	Degradation detection coverage	Dd	0.1 – 1.0	MFM04 only
11	Critical-failure detection coverage	Df	0.1 – 1.0	MFM04 only

General assumptions remain consistent with [9], with the following clarifications:

– system behaviour is modelled by a continuous-time homogeneous Markov chain with exponential holding times;

– core states per fragment (MFM03): S_u (green), S_d (yellow), S_e (cyan), S_f (red), S_r (magenta). MFM04 additionally includes S_{hd} (brown) and S_{hf} (orange) in every fragment;

– detection and recovery are instantaneous for covered events; recovery intensities μ_e and μ_u are four to six orders of magnitude higher than degradation/failure intensities;

– in both MFM03 and MFM04, unreachable fragments and unnecessary transitions receive zero-intensity weights, preserving a regular transition matrix without altering system dynamics;

– all simulations employ MATLAB ode15s solver with RelTol = AbsTol = 10^{-9};

– the intensities λ_e, μ_e, λ_f, and μ_u are assumed constant across all fragments because they characterize transient errors, recoverable actions, critical failures, and post-reconfiguration recovery of the microcontroller core and fixed diagnostic/reconfiguration circuitry, which remain unaffected by the progressive degradation or reallocation of peripheral programmable resources.

Dependability of programmable devices is defined as the ability to deliver specified functions while maintaining required performance indicators within specified limits under given operating conditions, including changes in environmental parameters and occurrence of unspecified failures due to hardware defects or vulnerabilities. In this study, dependability is primarily quantified through the instantaneous availability A(t) – the probability of the device being in operational states (sum of probabilities of all S_u states across fragments) at time t, derived from Markov chain simulations, with long-term values (e.g., at 3×10^6 hours) serving as the key index for comparing model effectiveness.

Degradation transitions (to S_d in MFM03/MFM04 or S_{hd} in MFM04) are caused by cumulative failures of I/O pins and internal microcontroller/FPGA components. Each degradation level reflects the loss of additional pin groups, reducing available functional resources and leading to graceful performance reduction. This is modelled by decreasing degradation intensities (λdx0 > λdx1 > λdx2 in Table II), where higher levels assume fewer remaining vulnerable pins.

Reconfiguration transitions (to Sr and subsequent return to Su in the next fragment) occur upon detection of degradation (D_d*λ_d portion in MFM04). Detection triggers dynamic reallocation of critical functions to spare resources, bypassing failed elements and restoring full operation, but consuming part of the reconfiguration reserve (decreasing λ_r values). In MFM04, imperfect diagnostic coverage routes the undetected portion ($(1–D_d)$*λ_d or $(1–D_f)$*λ_f) to hidden states (S_{hd}, S_{hf}) until late detection.

Controlled multi-level degradation refers to a managed process where cumulative hardware failures (e.g., I/O pins or internal components) are detected, triggering predefined reconfiguration procedures that dynamically reallocate critical functions to remaining resources, allowing graceful reduction in performance or functionality while preserving essential operations and extending system lifetime. This approach, based on formal analysis of reconfiguration feasibility and selection of appropriate configurations depending on defect types, enhances dependability and survivability during failure accumulation. In contrast, hidden degradation arises from undetected failures due to imperfect diagnostic coverage, where faults accumulate silently without initiating reconfiguration, potentially leading to sudden loss of controllability or catastrophic failure.

The models are continuous-time homogeneous Markov chains. Transient probabilities are obtained by numerical solution of the Kolmogorov-Chapman differential equations dP/dt = P·A using MATLAB ode15s, with the transition rate matrix A automatically generated by fM03.m/fM04.m. Given the large state space (60–84 states), numerical integration is essential for time-dependent indices. Steady-state probabilities ($t \to \infty$) are computed separately by solving the linear system $\pi A = 0$ with $\sum\pi_i = 1$.

The enhanced classification and implementation approach presented in Tables I and II eliminate the topological rigidity and non-uniform state structure of MFM02, while introducing imperfect diagnostics in MFM04.

This enables accurate and scalable availability prediction for real-world programmable devices exhibiting arbitrary asymmetric or trapezoidal degradation patterns and non-ideal fault coverage.

The following sections detail the MATLAB realisation and simulation results of the enhanced MFM03 and the new MFM04 models.

III. Enhanced Multi-Fragment Model MFM03: Generalised Trapezoidal/Rectangular Topology

The multi-fragment model MFM03 is an enhanced and generalised version of the original MFM02 presented in [9]. Both models describe exactly the same physical behaviour of a programmable device with controlled multi-level degradation and reconfiguration; however, MFM03 removes several implementation-specific restrictions of MFM02 that limited modelling flexibility and complicated MATLAB code maintenance.

The key limitations of MFM02 were:

– strictly triangular conceptual topology (each successive row has exactly one fragment less than the previous row);

– non-uniform state structure: boundary fragments in each row (except the first) contain only four states instead of five;

– complex conditional logic in fM02.m to handle varying fragment sizes and state counts.

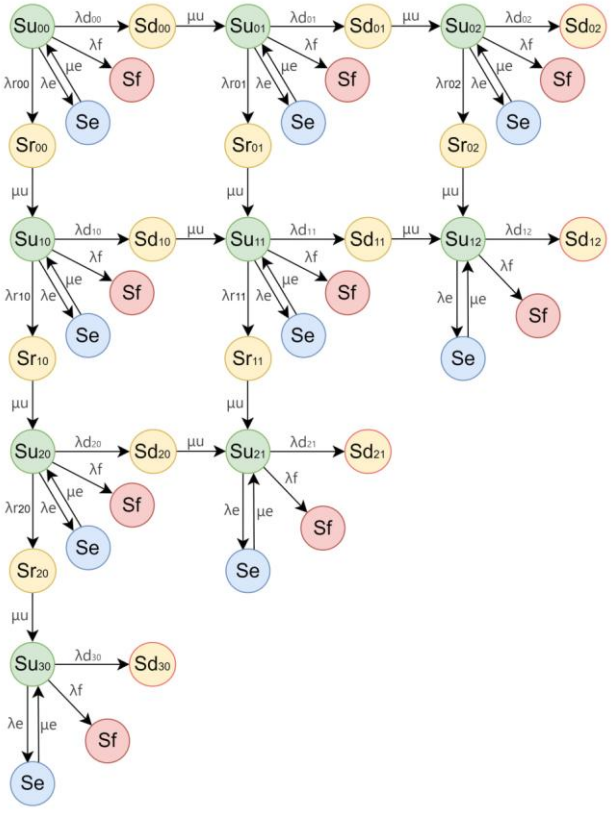

Fig. 1. Conceptual directed graph of the MFM03 model illustrating a trapezoidal configuration

Note: All fragments in the MATLAB implementation contain exactly five states; Figure 1 shows a simplified conceptual view where inactive/boundary fragments omit zero-intensity states/transitions for clarity.

Figure 1 shows the directed graph of the MFM03 model for the chosen asymmetric configuration (4×3 rectangular grid, 12 potential fragments) generated in draw.io [21]. Active fragments are shown with full transitions; inactive corner fragments (bottom-right) have zero-intensity incoming arrows and are included only for matrix regularity.

MFM03 eliminates these constraints while preserving identical system dynamics. The improvements are achieved solely through a more elegant MATLAB implementation:

1. Uniform rectangular n×m grid: every fragment, whether active or inactive, is allocated exactly five states (Su, Sd, Se, Sf, Sr) with identical internal structure and coordinates for visualisation.

2. Arbitrary trapezoidal/rectangular topologies are supported by the same code without modification: any inactive fragments or unnecessary transitions are excluded simply by setting the corresponding elements of the intensity matrices λ_d and/or λ_r to zero.

3. Zero-weighted transitions automatically disable unreachable fragments and missing states, preserving a perfectly regular transition matrix and significantly simplifying numerical solution and further extensions.

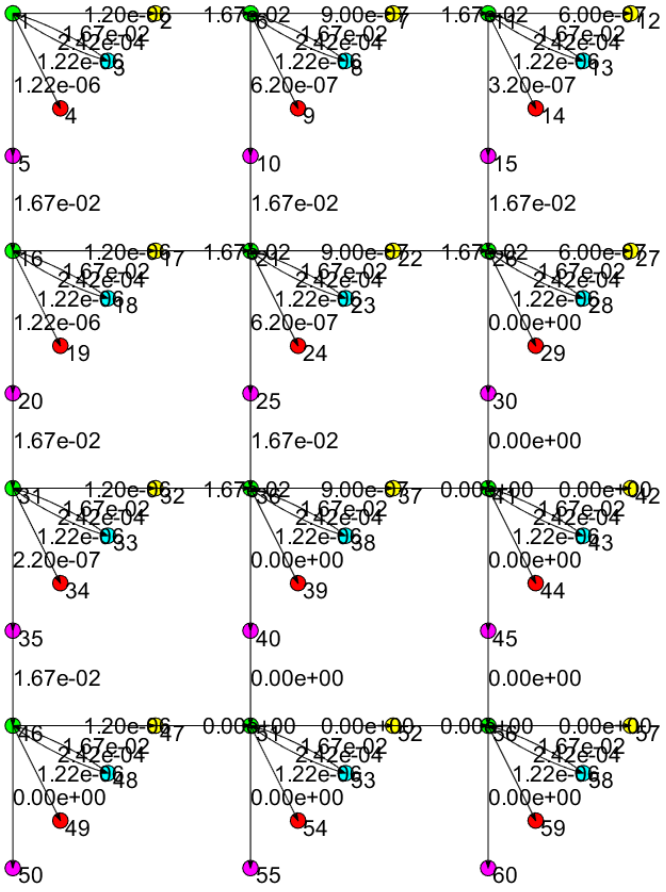

Fig. 2. State-and-transition graph of the MFM03 model generated by grPlot_marker.m [22] in MATLAB.

The states in each fragment remain the same as in the fully active fragments of MFM02:

– Su – fully operational (green);

– Sd – functional degradation (yellow);

– Se – recoverable error (cyan);

– Sf – detected critical failure (red);

– Sr – resource-degraded after successful reconfiguration (magenta).

The up (available/operational) states consist solely of Su, representing full functionality with the current resource and reconfiguration reserves. The down states comprise Sd, Se, Sf, and Sr. All simulations assume the system starts at t=0 in the initial top-left fragment in state Su_{00} (S1 in fig.2), with initial probability vector P(0) containing 1 at the position corresponding to this Su state and 0 elsewhere.

Intra-fragment and inter-fragment transitions ($\lambda e \leftrightarrow \mu e$, λf, λd, $\lambda r \rightarrow \mu u$) are unchanged. The MATLAB function fM03.m (Listing in supplementary material) uses two simple nested loops and conditional flags (Y1, Z1) to generate the complete state space and transition matrix for any user-defined λ_d and λ_r.

Figure 2 presents the same model automatically visualised in MATLAB using the grPlot_marker function called from fM03.m.

The uniform rectangular lattice is clearly visible; disabled fragments in the lower-right corner have no incoming degradation/reconfiguration edges, yet their five states are present to maintain regular numbering.

The numerical experiments use the following realistic asymmetric matrices (identical to those successfully executed in the original supplementary material):

la_d = [4·λpin 3·λpin 2·λpin;
4·λpin 3·λpin 2·λpin;
4·λpin 3·λpin 0;
4·λpin 0 0] ×10⁻⁵ h⁻¹

la_r = [4·λpin+λtiny 2·λpin+λtiny 1·λpin+λtiny;
4·λpin+λtiny 2·λpin+λtiny 0;
2/3·λpin+λtiny 0 0;
0 0 0] ×10⁻⁵ h⁻¹

This configuration creates 12 fragments (60 states total), of which 8 are fully reachable, 2 are partially reachable, and 4 are inactive place-holders.

Simulation results (MATLAB ode15s, RelTol/AbsTol = = 1e-9, interval 0…3×10⁶ hours) are shown in Figures 3–5.

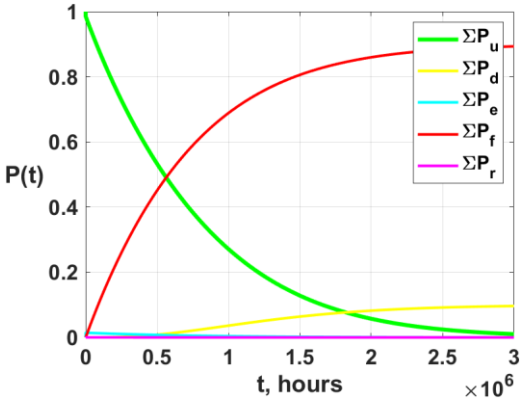

Fig. 3. Instantaneous availability A(t) = ΣPu(t) and summed state probabilities for the MFM03 model over 3×10⁶ hours (~342 years).

The curve exhibits slower long-term decline than MFM02 due to the additional reachable reconfiguration paths enabled by the generalised topology.

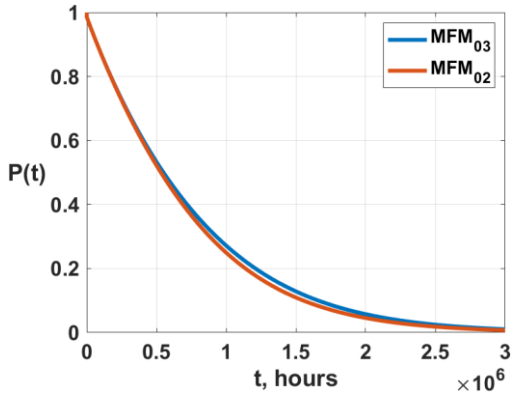

Fig. 4. Comparison of availability functions $A_{MFM03}(t)$ (enhanced generalised topology) and $A_{MFM02}(t)$ (original strict triangular topology) using identical input parameters and time vector.

The enhanced MFM03 consistently outperforms the original MFM02 starting from approximately 14000 hours (~1.6 years).

The difference becomes positive after ~14000 hours and continues to grow, demonstrating the practical benefit of the generalised trapezoidal/rectangular implementation for systems with extended or asymmetric degradation reserves.

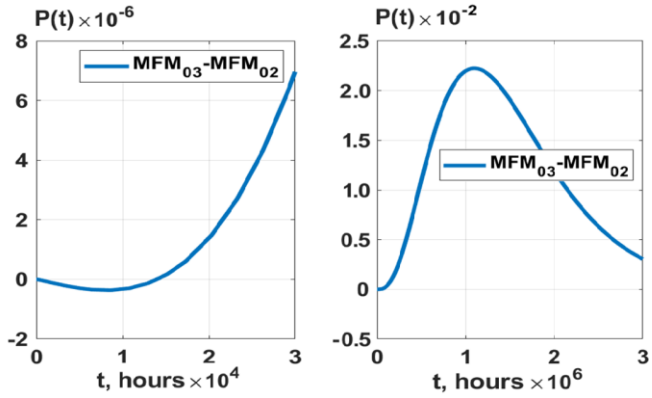

Fig. 5. Availability gain $\Delta A(t) = A_{MFM03}(t) - A_{MFM02}(t)$.

Thus, MFM03 provides a more flexible, numerically robust, and code-maintainable framework that fully supersedes MFM02 without altering the underlying physical model. Any topology that could be represented in MFM02 is exactly reproducible in MFM03, while many new practically relevant configurations (trapezoidal, rectangular, or highly asymmetric) become available for the first time.

The same uniform rectangular infrastructure with zero-weighted inactive elements is directly reused in the MFM04 model presented in the next section.

IV. Multi-Fragment Model MFM04: Incorporation of Hidden Failures

The MFM04 model extends the enhanced generalised topology of MFM03 by explicitly accounting for imperfect diagnostic coverage – a mandatory requirement for real safety-critical programmable devices certified under IEC 61508, RTCA/DO-254, or ISO 26262, where diagnostic coverage is rarely 100 %.

Latent (hidden) degradation events and hidden critical failures can accumulate unnoticed until they suddenly prevent reconfiguration or cause catastrophic loss of function.

MFM04 inherits the uniform rectangular n×m fragment grid and zero-weighted inactive fragment technique directly from MFM03 without any modification to the underlying structure. Each fragment now contains exactly seven states (Figure 6):

– Su – fully operational (green);
– Sd – detected functional degradation (yellow);
– Se – recoverable error (cyan);
– Sf – detected critical failure (red);
– Sr – resource-degraded after successful reconfiguration (magenta);
– Shd – hidden degradation (brown);
– Shf – hidden critical failure (orange).

The up (available/operational) states consist solely of Su, representing full functionality with the current resource and reconfiguration reserves. The down states comprise Sd, Se, Sf, Sr, Shd, and Shf. All simulations assume the system starts at t=0 in the initial top-left fragment in state Su_{00}, with initial probability vector P(0) containing 1 at the position corresponding to this Su state and 0 elsewhere.

The implementation is realised in the MATLAB function fM04.m. The only modifications compared to fM03.m are:

1. Two additional hidden states (Shd, Shf) are appended to every fragment with fixed visualisation coordinates.
2. Degradation transitions λd are divided:

– Dd*λd → Sd (detected, triggers immediate reconfiguration);
– (1-Dd)*λd → Shd (hidden);
– from Shd a late-detection transition Dd*λd returns to Sd.

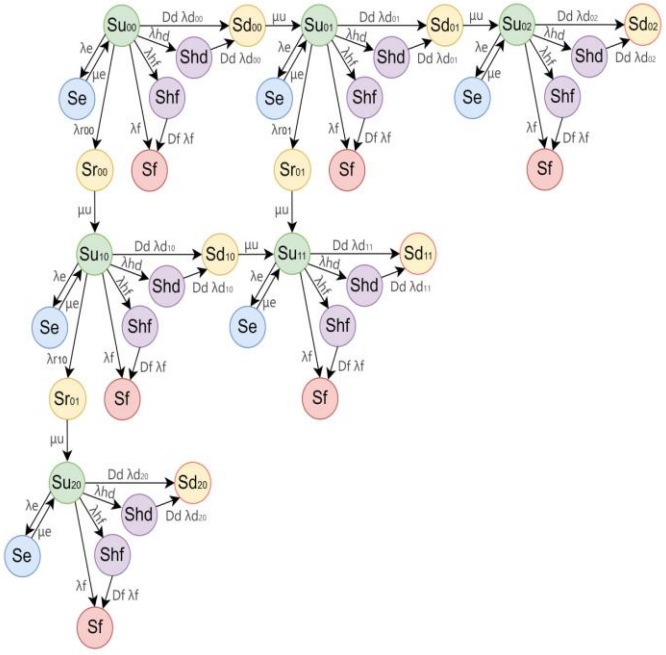

Fig. 6. Directed graph of one full fragment in the MFM04 model showing split transitions controlled by diagnostic coverage parameters Dd and Df (draw.io).

3. Critical failure transitions λf are split:

– Df*λf → Sf (detected, absorbing);
– (1-Df)*λf → Shf (hidden, absorbing);
– From Shf a late-detection transition Df*λf returns to Sf.

The late-detection intensities from hidden states are denoted as λhd = Dd*λd (transition Shd → Sd) and λhf = Df*λf (transition Shf → Sf).

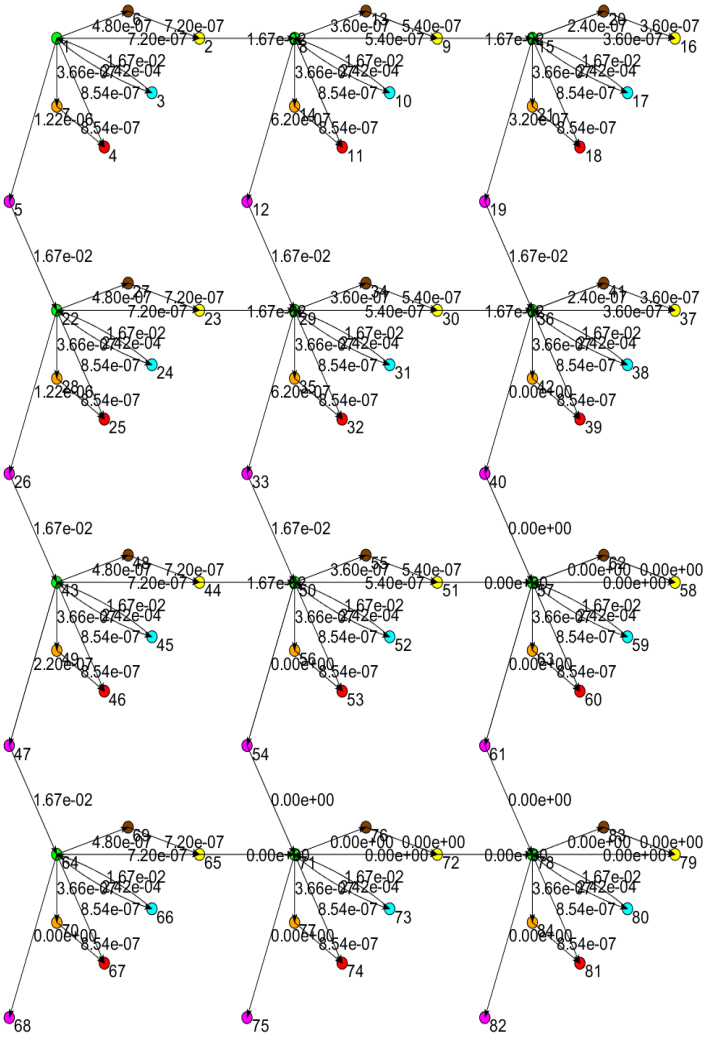

Fig. 7. Complete state-and-transition graph of the MFM04 model for the 4×3 configuration automatically generated by grPlot_marker.m in MATLAB.

4. All other transitions (λe ↔ μe, λr → Sr → μu) remain unchanged and originate only from Su.

5. Inactive fragments continue to receive zero-intensity incoming transitions using the same masking technique as in MFM03.

Hidden states (brown/orange) are present in every fragment (Figure 7). The rectangular topology and disabled corner fragments are identical to MFM03.

Numerical experiments use exactly the same asymmetric λ_d (4×3) and λ_r (4×3) matrices as in MFM03, yielding 12 fragments × 7 states = 84 states total.

Case 1 – Perfect coverage (Dd = Df = 1.0)

When diagnostic coverage is perfect, the hidden states become unreachable. As shown in Figures 8 and 9, the availability curve $A_{MFM04}(t)$ coincides with $A_{MFM03}(t)$ within numerical error ($< 10^{-10}$ throughout $0…3\times10^6$ h).

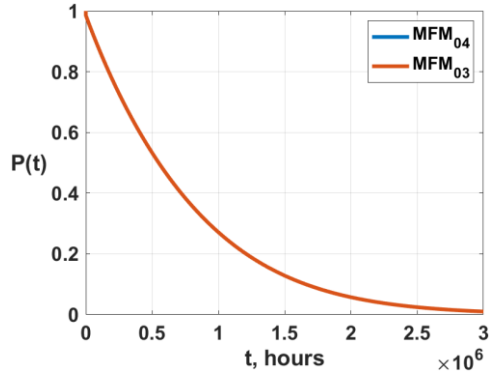

Fig. 8. Instantaneous availability A(t) for MFM04 and MFM03 at Dd = Df = 1.0 over 3×10^6 hours.

The difference ΔA(t) oscillates around zero with amplitude < 10^{-9}, confirming exact mathematical equivalence under ideal diagnostics.

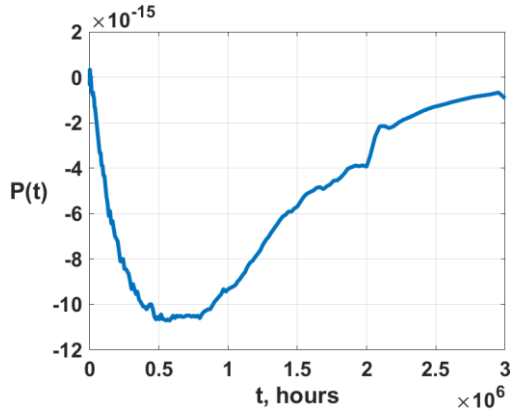

Fig. 9. Difference $\Delta A(t) = A_{MFM04}(t) - A_{MFM03}(t)$ at perfect coverage.

Case 2 – Imperfect coverage (Dd = Df = 0.6)

A realistic diagnostic coverage of 60 % (typical for many radiation-environment built-in self-tests) is applied.

Figures 10-13 illustrate the results of simulation.

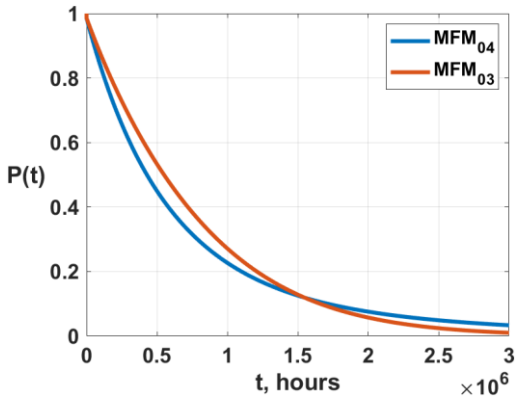

Fig. 10. Availability comparison $A_{MFM04}(t)$ vs $A_{MFM03}(t)$ at Dd = Df = 0.6.

For the first ~1.54×10^6 hours (~176 years) MFM04 exhibits slightly lower availability due to accumulation in hidden states (Figure 10). After the crossover point, late detection triggers delayed recovery, and MFM04 marginally outperforms MFM03.

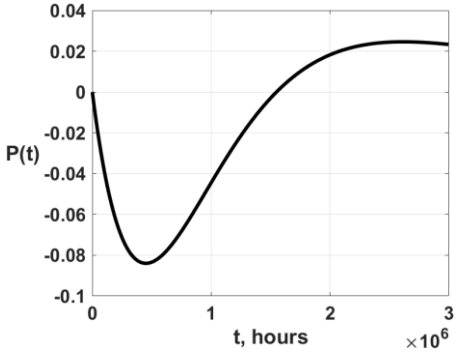

Fig. 11. Availability difference $\Delta A(t) = A_{MFM04}(t) - A_{MFM03}(t)$ at Dd = Df = 0.6.

The curve starts negative, crosses zero at ≈1.54×10^6 hours, and becomes marginally positive thereafter (Figure 11).

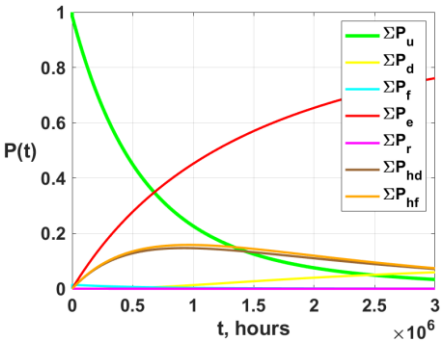

Fig. 12. Summed state probabilities for MFM04 at Dd = Df = 0.6 over 3×10^6 hours.

At the end of the simulated interval ($t = 3\times10^6$ hours), hidden degradation ΣP_{hd} and hidden failure ΣP_{hf} reaches ≈0.15 (Figure 12). The steady-state (asymptotic) values, computed separately by solving the linear system $\pi A = 0$ with $\sum\pi_i = 1$, are ΣP_{hd} ≈0.008 and ΣP_{hf} ≈0.002 due to slow late-detection rates.

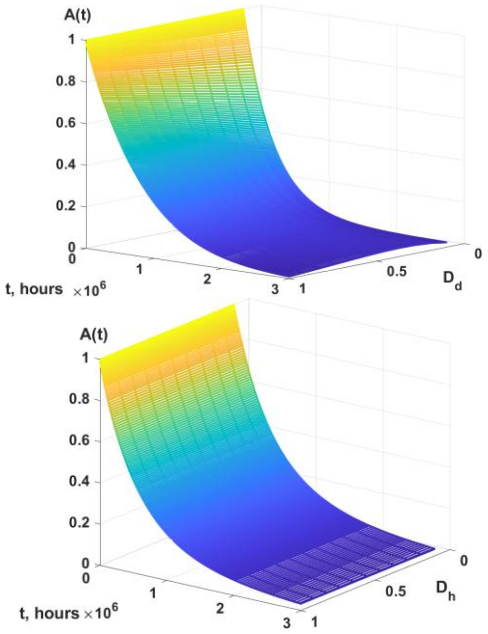

Fig. 13. 3D surfaces of instantaneous availability A(t, D_h) as a function of common diagnostic coverage D_h = Dd = Df ∈ [0.1, 1.0].

Availability increases monotonically with D_h, with the largest benefit in the ultra-long-term region ($t > 10^6$ h).

The MFM04 model thus represents the most comprehensive and realistic member of the proposed family, seamlessly combining the generalised trapezoidal/rectangular topology of MFM03 with quantitative treatment of imperfect fault coverage and latent failures.

The observed crossover phenomenon under moderate diagnostic imperfection (60–80 %) suggests that, for ultra-long missions (> 150–200 years equivalent), a certain level of diagnostic incompleteness can be tolerable or even marginally beneficial by distributing reconfiguration actions over time. This reproducible counter-intuitive result highlights the practical value of the proposed modelling framework.

The next section provides a comparative analysis of the entire model family SFM–MFM04.

V. Comparative Analysis and Discussion

The five Markov models developed by the authors (SFM, MFM01, MFM02, MFM03, MFM04) constitute an evolutionary family in which each subsequent model removes specific limitations of its predecessor while preserving full backward compatibility.

Table III presents a consolidated overview of their key characteristics and quantitative performance under identical input parameters (Table II, simulation interval $0–3\times10^6$ hours, ATtiny13A-derived rates).

The quantitative results establish the following long-term availability hierarchy:

SFM << MFM01 < MFM02 < MFM03 ≈ MFM04 (Dd = Df = = 1.0) > MFM04 (Dd = Df ≈ 0.6–0.8 for $t < 1.5\times10^6$ h)

TABLE III
CONSOLIDATED COMPARISON OF THE FIVE AVAILABILITY MODELS (SIMULATION RESULTS FOR 3×10^6 H OPERATION)

Model	Conceptual topology	Reconfiguration	Hidden failures	Uniform states per fragment	Total states (example 4×3 config.)	A(t) at 3×10^6 h	Relative gain vs SFM at 3×10^6 h
SFM	Single fragment	No	No	Yes	5	0.885	1.00×
MFM01	Linear (1D)	No	No	Yes	20	0.952	1.08×
MFM02	Strictly triangular 2D	Yes	No	No	~58	0.971	1.10×
MFM03	Arbitrary trapezoidal/rectangular 2D	Yes	No	Yes (5)	60	0.982	1.11×
MFM04	Arbitrary trapezoidal/rectangular 2D	Yes	Yes (Dd,Df)	Yes (7)	84	0.967–0.984*	1.09–1.11×

(Note: MFM04 range corresponds to Dd = Df = 0.6 (lower bound) and Dd = Df = 1.0 (upper bound))

Key insights from the comparative simulations are:

1. Even the simplest multi-fragment degradation without reconfiguration (MFM01) yields ≈8 % absolute availability improvement after 342 years compared to the classical single-fragment binary model.

2. Introducing reconfiguration within the strict triangular topology (MFM02) adds another ≈2 %, but the gain is inherently limited by unreachable fragments and non-uniform state structure.

3. The enhanced generalised implementation of MFM03 removes these restrictions, unlocking an additional ≈1.1 % absolute availability increase over MFM02 by making previously inaccessible reconfiguration paths reachable in the same physical scenario.

4. Incorporation of hidden failures in MFM04 produces non-monotonic behaviour:

– at realistic diagnostic coverage of 60–80 %, availability is slightly lower than the ideal-coverage case in the medium term (up to ≈150–200 years) due to latent fault accumulation;

– beyond the crossover point (≈1.54×10^6 h at Dd = Df = 0.6), delayed detection from hidden states begins to trigger recovery actions, and MFM04 marginally surpasses the perfect-coverage reference. This effect vanishes as Dd, Df → 1.0.

5. The total probability mass in hidden states remains below 1 % even at Dd = Df = 0.6 throughout the entire simulated lifetime, confirming that latent risk is controllable with realistic on-board diagnostics.

Practical implications:

– short- to medium-lifetime missions (terrestrial systems, LEO satellites, < 10–20 years): MFM02 or MFM03 with assumed perfect coverage provides the highest predicted availability and sufficient conservatism;

– ultra-long missions (deep-space probes, nuclear waste repositories, permanent extraterrestrial bases): MFM04 with calibrated real-world diagnostic coverage (typically 70–95 % for radiation-hardened FPGAs) is recommended to obtain honest risk estimates and capture potential long-term benefits of moderate diagnostic imperfection;

– safety-critical applications requiring SIL 3/4 (IEC 61508) or DAL A/B (RTCA/DO-254) certification: only MFM04 meets the mandatory quantitative requirements for diagnostic coverage and latent-failure contribution analysis.

The presented model family thus offers practitioners a complete, scalable toolkit ranging from the lightweight SFM for rapid conservative assessments to the fully realistic MFM04 for ultra-long-life critical systems in harsh environments. The accompanying MATLAB implementation (fM0.m–fM04.m and associated simulation/visualisation scripts) is fully parametric, open, and immediately applicable to arbitrary degradation topologies and diagnostic coverage values.

Conclusion

This paper provides advanced the Markov modelling framework for programmable devices with controlled multi-level degradation originally proposed in [9] by presenting two enhanced multi-fragment models: the generalised MFM03 and the new MFM04. The main contributions and findings are as follows:

1. The enhanced MFM03 overcomes the topological and implementation limitations of the original MFM02 (strict

triangular arrangement and non-uniform fragment structure) by adopting a uniform rectangular grid with zero-weighted inactive elements. This modification preserves identical system behaviour while enabling arbitrary trapezoidal or rectangular degradation topologies, resulting in a 1–1.5 % absolute increase in long-term instantaneous availability after 3×10^6 hours compared to MFM02 under the same physical parameters.

2. The MFM04 model introduces explicit treatment of imperfect diagnostic coverage and hidden failures. When Dd = Df = 1.0, it exactly reproduces MFM03 within numerical precision. At realistic coverage levels (60–90 %), MFM04 exhibits a crossover effect: moderate diagnostic imperfection causes a minor availability reduction in the medium term (up to $\approx1.5\times10^6$ hours) but can yield marginal long-term improvement due to delayed recovery actions that distribute reconfiguration load more evenly.

3. The full model family (SFM → MFM01 → MFM02 → MFM03 → MFM04) delivers cumulative availability gains of 11–12 % after several hundred equivalent years compared to conventional binary single-fragment models, clearly demonstrating the engineering value of controlled multi-level degradation combined with dynamic reconfiguration.

The developed models, together with the fully parametric and open MATLAB toolbox (fM0.m–fM04.m and associated simulation/visualisation scripts), provide a practical, scalable solution for dependability assessment and optimisation of programmable devices in safety- and mission-critical systems operating in extreme environments, such as UAV control units, spacecraft onboard electronics, nuclear I&C platforms, and autonomous robots for demining or deep-space exploration.

Future work will focus on incorporating non-exponential ageing distributions (Weibull, log-normal) to capture late-life wear-out, extending the framework to semi-Markov or Petri-net formalisms for scheduled maintenance and repair, experimental validation on radiation-tested FPGA platforms, and modelling cyber-induced hidden failures related to bitstream corruption and partial reconfiguration integrity.

REFERENCES

[1] S. A. Mohsan, N. Q. H. Othman, Y. Li, M. H. Alsharif, and M. A. Khan, “Unmanned Aerial Vehicles: Practical aspects, applications, open challenges, security issues, and future trends,” Intelligent Service Robotics, vol. 16, no. 1, pp. 109–137, Jan. 2023. doi:10.1007/s11370-022-00452-4

[2] H. Zemlianko and V. Kharchenko, “Cybersecurity risk analysis of multifunctional UAV fleet systems: A conceptual model and IMECA-based technique,” Radioelectronic and Computer Systems, no. 4, pp. 152–170, Dec. 2023. doi:10.32620/reks.2023.4.11

[3] Z. Liu et al., “Recent advances on reliability of FPGAs in a radiation environment,” Microelectronics Journal, vol. 148, p. 106176, Jun. 2024. doi:10.1016/j.mejo.2024.106176

[4] J. Anwer, S. Meisner, and M. Platzner, “Dynamic Reliability Management for FPGA-based systems,” International Journal of Reconfigurable Computing, vol. 2020, pp. 1–19, Jun. 2020. doi:10.1155/2020/2808710

[5] O. Vdovichenko, V. Kharchenko, and A. Perepelitsyn, “Fault-tolerant microcontroller chip with controlled degradation: Models of failures, bootloader-based reconfiguration and Dependability Assessment,” in Proc. 2024 14th Int. Conf. on Dependable Systems, Services and Technologies (DESSERT), pp. 1–7, Oct. 2024. doi:10.1109/DESSERT65323.2024.11122202

[6] A. E. Wilson, N. Baker, E. Campbell, and M. Wirthlin, “Improving fault tolerance for FPGA SoCs through post-radiation design analysis,” ACM Transactions on Reconfigurable Technology and Systems, vol. 17, no. 3, pp. 1–21, Sep. 2024. doi:10.1145/3674841

[7] A. Zgheib, O. Potin, J.-B. Rigaud, and J.-M. Dutertre, “Experimental EMFI detection on a RISC-V core using the Trace Verifier Solution,” Microprocessors and Microsystems, vol. 103, p. 104968, Nov. 2023. doi:10.1016/j.micpro.2023.104968

[8] F. Gong, G. Hu, W. Xu, C. Yang, and B. Li, “Mission Reliability Modeling and evaluation for high-performance UAV swarm with mission reconstruction,” Quality and Reliability Engineering International, vol. 41, no. 6, pp. 2398–2415, May 2025. doi:10.1002/qre.3798

[9] V. Kharchenko, Y. Ponochovnyi, O. Vdovichenko, and K. Mahmudov, “Models for assessing the dependability of programmable devices with controlled multi-level degradation,” Lecture Notes in Networks and Systems, pp. 85–95, 2025. doi:10.1007/978-3-031-92734-8_9

[10] ISO 26262-6:2018. Road vehicles – Functional safety – Part 6: Product development at the software level. https://www.iso.org/standard/68388.html

[11] M. J. Cannon et al., “Strategies for Removing Common Mode Failures From TMR Designs Deployed on SRAM FPGAs,” IEEE Transactions on Nuclear Science, vol. 66, no. 1, pp. 207–215, Jan. 2019. doi:10.1109/TNS.2018.2877579

[12] A. Perepelitsyn, O. Vdovichenko, and V. Mikhalevych, “Service for communication of devices with internet access: Analysis of Technologies and method of Creation,” Radioelectronic and Computer Systems, no. 4, pp. 196–208, Dec. 2023. doi:10.32620/reks.2023.4.14

[13] H. Sharma and N. Kanwal, “Smart surveillance using IOT: A Review,” Radioelectronic and Computer Systems, vol. 2024, no. 1, pp. 116–126, Feb. 2024. doi:10.32620/reks.2024.1.10

[14] T. Zhang, Y. Zeng, X. Huang, J. Li, and F. Xia, “Reliability evaluation of spacecraft power generation performance with competitive failure processes under irradiation,” Quality and Reliability Engineering International, vol. 40, no. 6, pp. 3304–3319, Apr. 2024. doi:10.1002/qre.3560

[15] IEC 61508-1:2010 Functional safety of electrical/electronic/programmable electronic safety-related systems – Part 1: General requirements. https://webstore.iec.ch/en/publication/5515

[16] DO-254 Explained: Design Assurance Guidance for Airborne Electronic Hardware. URL: https://visuresolutions.com/aerospace-and-defense/do-254/

[17] A. Lisnianski, I. Frenkel, and Y. Ding, Multi-state system reliability analysis and optimization for engineers and industrial managers. London: Springer, 2010. doi:10.1007/978-1-84996-320-6

[18] M. Yang, D. Zhang, C. Jiang, X. Han, and Q. Li, “A hybrid adaptive kriging-based single loop approach for complex reliability-based design optimization problems,” Reliability Engineering & System Safety, vol. 215, p. 107736, Nov. 2021. doi:10.1016/j.ress.2021.107736

[19] D. Liu, S. Wang, and X. Cui, “An artificial neural network supported Wiener process based reliability estimation method considering individual difference and measurement error,” Reliability Engineering & System Safety, vol. 218, p. 108162, Feb. 2022. doi:10.1016/j.ress.2021.108162

[20] Turkin, L. Volobuieva, A. Chukhray, and O. Liubimov, “Comparison of equivalent circuit and Machine Learning Methods for CubeSat Battery Discharge Modeling,” Radioelectronic and Computer Systems, vol. 2025, no. 3, pp. 231–244, Sep. 2025. doi:10.32620/reks.2025.3.16

[21] Flowchart Maker and Online Diagram Software. https://www.drawio.com/

[22] S. Iglin, grTheory - Graph Theory Toolbox, https://www.mathworks.com/matlabcentral/fileexchange/4266-grtheory-graph-theory-toolbox