Proceedings of the 35th European Safety and Reliability & the 33rd Society for Risk Analysis Europe Conference
Edited by Eirik Bjorheim Abrahamsen, Terje Aven, Frederic Bouder, Roger Flage, Marja Ylönen
 Published by Research Publishing, Singapore.
doi: 10.3850/978-981-94-3281-3_ESREL-SRA-E2025-P5334-cd

A Topology-Based Method for Quantifying and Evaluating the System Structural Fault Tolerance Capability

Yi-Yang Shangguan

School of Reliability and Systems Engineering, Beihang University, China. E-mail: shangguanYy@buaa.edu.cn

Xiao-Yang Li

School of Reliability and Systems Engineering, Beihang University, China. E-mail: leexy@buaa.edu.cn

Wen-Bin Chen

School of Reliability and Systems Engineering, Beihang University, China. E-mail: chenwenbin@buaa.edu.cn

Shi-Wen Zhang

School of Reliability and Systems Engineering, Beihang University, China. E-mail: SY2214122@buaa.edu.cn

Rui Kang

School of Reliability and Systems Engineering, Beihang University, China. E-mail: kangrui@buaa.edu.cn

Abstracts
In modern complex systems, the capability to sustain operations despite component failures is defined as the fault tolerance capability. To ensure system reliability, it is essential to quantify and evaluate the fault tolerance capability systematically. However, existing research primarily focuses on failure propagation and lacks the consideration of critical thresholds for fault tolerance, leading to inadequate guidance for fault-tolerant system. To tackle these challenges, this paper proposes a topology-based method for quantifying and evaluating the system structural fault tolerance. By modelling systems as the directed graph based on the functional logic, we introduce the all functional paths (AFP) to measure redundancy through valid input-to-output paths. The structural fault tolerance index is derived by normalizing AFP values under fault conditions against a fault-free baseline, which is applicable to both single-input single-output (SISO) and multi-input multi-output (MIMO) systems. Additionally, we define the fault tolerance threshold based on the minimum redundancy requirement and introduce the margin to assess compliance. A case study on the redundant flywheel system (RFS) demonstrates the method's feasibility, and we calculate the structural fault tolerance index and the corresponding fault tolerance margin of the RFS in each state. The proposed framework bridges component-level faults to system-level functionality, offering actionable insights for evaluating and improving redundant system.

Keywords: Fault tolerance, structural diagram model, margin, redundant flywheel system.

1. Introduction

In today's rapidly evolving technological landscape, complex systems in fields such as aerospace (Bennett et al. 2012), automotive (Manzone, Pincetti, and Costantini 2001), industrial control (Campelo et al. 1999), and computer networks (Bohacek et al. 2007) increasingly demand both high reliability and robust fault tolerance. The capability to maintain the essential functionality in the event of partial failures has become a fundamental priority. As the scale and complexity of such systems continue to grow, the probability of the failures increases, highlighting the importance of quantifying fault tolerance in a systematic way. This underscores the urgent need for systematic quantification of fault tolerance capability, which is the crucial requirement to guide

engineers in prioritizing design efforts and allocating resources effectively to ensure operational continuity.

To strengthen system reliability and fault tolerance capability, domain-specific research has produced diverse fault tolerance techniques. Computing systems employ techniques including multi-module redundancy (Jin et al. 2022), dynamic redundancy with fault detection (Krishna and Shin 1986), watchdog processors (Chen et al. 1994), *N*-version programming (Dugan and Lyu 1994), while mechanical systems utilize redundant architectures such as six-legged robots (Ning et al. 2019) and multi-joint manipulators (Gong, Li, and Zhang 2019). Though the above strategies are effective, they lack the unified metrics to evaluate the systemic impact for the fault tolerance capability.

Based on the above fault tolerance techniques, researchers have developed various quantitative methods to evaluate fault tolerance capability. Classical approaches, such as reliability block diagrams (RBD) (Bistouni and Jahanshahi 2014) and fault tree analysis (FTA) (Dugan, Bavuso, and Boyd 1992), rely on the reliability metrics like mean time to failure (MTTF) and mean time to recovery (MTTR) (Al-Kuwaiti, Kyriakopoulos, and Hussein 2009) as proxies for fault tolerance. Fault coverage (AMARI 2007), defined as the ratio of manageable faults to total possible faults, provides another evaluation perspective. For network-based systems, graph-theoretic methods are widely used, with connectivity metrics (e.g., *k*-vertex or *k*-edge connectivity) assessing fault endurance. For example, (Lei and Meng 2020) introduced a fault-tolerant Hamiltonian measure for hierarchical permutation graphs. Furthermore, in certain domains—such as data centers, fault tolerance is often equated with system resiliency, quantified using resiliency triangle areas (Li et al. 2021).

However, existing research primarily focuses on the failure logic of systems. For instance: RBD analyze failure behaviors at the component level; FTA models fault events propagating within the system; Markov processes capture state transitions from normal operation to failure; the network models rely on node failure propagation as their foundation. In essence, these methods are inherently failure-centric, making them inadequate for guiding fault-tolerant system design improvement. Furthermore, they lack the consideration of fault tolerance requirements and fail to quantify critical thresholds of fault tolerance capability.

To address these challenges, this paper proposes a topology-based method for quantifying and evaluating the system structural fault tolerance capability. We focus on the concept of the system state, determined by each component's operating status. Each component may have multiple fault modes and the system structure function maps these component states to the system-level state, thereby modeling complex fault scenarios. Further, the valid functional paths are analyzed and the total count of these paths, termed AFP (All Function Path), serves as the key measure of structural redundancy. By normalizing AFP values under fault conditions against the fault-free baseline, we obtain the structural fault tolerance index to both single-input single-output (SISO) and multi-input multi-output (MIMO) systems. Moreover, we propose the quantification method for the threshold of the structural fault tolerance, which can quantify the structural fault tolerance index under different requirements and use it as the structural fault tolerance threshold.

The remainder of this paper is organized as follows. Section 2 analyzes and describes the system state. Section 3 proposes the quantitative method for evaluating structural fault tolerance. Section 4 provides a case application. Finally, Section 5 summarizes the main work.

2. Description of the System State

For a component fault within a system, it can be considered that the component may exhibit multiple fault modes, and there exist certain logical relationships between the fault modes of different components. Furthermore, different fault modes of the same component may have different impacts on the system, while faults in different components may result in identical effects on the system, which is determined by the system's functional logic structure. Therefore, when describing the state of a component, it is necessary to consider its multiple fault modes and their transition relationships. Building upon this, a mapping from component states to system states can be established based on the system's functional logic structure, ultimately yielding the description of the system state.

For a system consisting of n_C components, each individual component c has a total of k_c possible failure modes. The state of this component δ_c can be defined as follows:

$$\delta_c = \begin{cases} 0 & \text{Normal} \\ 1 & \text{Failure mode 1} \\ \dots & \dots \\ k_c & \text{Failure mode } k_c \end{cases} \quad c = 1,2,...,n_c \quad (1)$$

where $\delta_c \in \{0,1,...,k_c\}$, which is the description of the component state; $\{0,1,...,k_c\}$ is the state space of this component.

The state of all components is represented by the vector $\left[\delta_1, \delta_2, ..., \delta_{n_c}\right]$, denoted as α_δ. The state of the system can be represented as the function of α_δ:

$$\boldsymbol{\delta} = f_s(\alpha_\delta) \quad (2)$$

where $\boldsymbol{\delta}$ is the system state; $f_s(\bullet)$ is the functions of the system's structure, which indicates the mapping of component state vector space to system state space, the set of all states of the system is defined as the state space of the system, denoted as $\mathbb{S}$.

The system state space is the Cartesian product of the state space of all components $\{0,1,...,k_1\} \times \{0,1,...,k_2\} \times ... \times \{0,1,...,k_{n_c}\}$, and the total number of states of the system is $\prod_{c=1}^{n_c}(k_c+1)$. As the number of components increases, the total number of system states grows exponentially, and the model suffers from the problem of state explosion. In fact, different components may have the same impact on the system, which leads to the same system state. Therefore, in practical applications, the system state needs to be simplified for the fault logic relationship between components to define a more accurate system state function.

As the degradation time increases, the state of the component changes, which further leads to a change in the state of the system, called state transfer. The transfer trajectory l_c of the state of a component c can be defined as:

$$l_c = \left\{\delta_c(\vec{t}) \in \{0,1,...,k_c\} \mid \vec{t} \in R^+\right\} \quad (3)$$

where $c = 1,2,...,n_c$.

Further, the state transfer trajectory of the system is:

$$l_{\text{sys}} = \left\{\boldsymbol{\delta}_{\vec{t}} = f_s\left(\alpha_\delta(\vec{t})\right) \in \mathbb{S} \mid \vec{t} \in R^+\right\} \quad (4)$$

In the finite and discrete cases, the value range of $\vec{t}$ is a set of degenerate time vectors $\left[\vec{t}_0, \vec{t}_1, ..., \vec{t}_M\right]$. The state transfer sequence of the components can be expressed:

$$l_c = \left[\delta_c(\vec{t}_0), \delta_c(\vec{t}_1), ..., \delta_c(\vec{t}_M)\right] \quad (5)$$

Further, the state transfer sequences of all components are combined into the matrix $L = \left[\delta_{ij}\right]_{n_c \times (M+1)}$, where $\delta_{ij} = \delta_i(\vec{t}_j)$ and $j = 0,1,...,M$. The cth row of this matrix is the state transfer sequence of the component c and the column vector is the input of the system state function $\alpha_\delta(\vec{t}_j)$ at the time $\vec{t}_j$, the corresponding system state is $\boldsymbol{\delta}(\vec{t}_j) = f_s\left(\alpha_\delta(\vec{t}_j)\right)$, and the state transfer sequence of the system can be obtained, denoted as:

$$l_{\text{sys}} = \left[\boldsymbol{\delta}(\vec{t}_0), \boldsymbol{\delta}(\vec{t}_1), ..., \boldsymbol{\delta}(\vec{t}_M)\right] \quad (6)$$

clearly, when L is determined, the transfer trajectory of the system is also uniquely determined. Thus, the transfer sequence of the system state is a function of the transfer sequences of all components.

3. Quantification of the System Structural Fault Tolerance Capability

This section is based on the description of system states and employs the topological relationships of graph theory to describe the system's structure. From the perspective of the system's functional implementation process, the model of the system structure can be constructed. Furthermore, the number of paths in graph can be used to measure the redundancy of the system structure, thereby establishing the quantitative model for the system's fault tolerance.

3.1. *Modelling of system structural diagram*

From a process-oriented perspective, a system receives inputs from its environment, processes them internally, and produces outputs as feedback. When the output meets the functional object requirements of the system, the system can realise the function. The system structure R is defined as the totality of material, informational, and energetic interactions among the components. This structure can be modelled

as a directed graph, where nodes represent component functions and edges denote their interactions.

Based on the system state $\boldsymbol{\delta}$ obtained in Section 2, the input-output relationships and component interactions of the system can be analysed. Further, a directed graph $G_{\boldsymbol{\delta}}(V,E)$ can be constructed, which is referred to as the structural graph of the system in state $\boldsymbol{\delta}$. It can be denoted as $V=\left\{s,\mathbf{c}_1,\mathbf{c}_2...,\mathbf{c}_{n_c},e\right\}$, where s is the input of the system and e is the output of the system. c_i is the set of points represented by the ith component. The elements in E represent the interactions between components, indicating the flow of material, energy, and information between different segments.

For a specific system, the steps for constructing a structural graph are as follows:

1) Analyze the system's component composition and functional principles.
2) Decompose the functions of each component.
3) Consider the functional logic between stages (e.g., selecting k out of n), and introduce virtual nodes as needed to ensure logical consistency.
4) Analyze the inputs and outputs of each functional stage to determine the edges of the graph.
5) Set the starting points of all system input edges as $s_1,s_2,...$ and the endpoints of all system output edges as $e_1,e_2,...$.

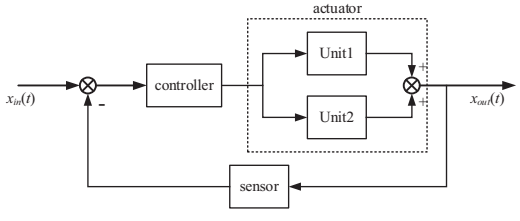

Fig. 1. Block diagram of the functional principle of the dual-redundancy negative feedback control system.

The structural graph proposed in this paper is derived from the system's functional logic, enabling it to assess the integrity of the system's functional logic. Taking a common dual-redundancy negative feedback control system as an example, its functional schematic is illustrated in Fig. 1. The system's components can be categorized into the controller, dual-redundancy actuator unit, and sensor.

The functional decomposition of the system's components reveals its primary functional modules: (1) the *controller*, which computes errors based on system inputs and outputs and generates control signals; (2) the *sensor*, which measures the actual output and feeds it back to the controller; (3) and the redundant actuator unit, comprising *actuator*1 and *actuator*2, which convert control signals into actual outputs. The two actuators operate in parallel, functioning independently without mutual influence, and each can independently provide output to the system. Based on this, the system's structural graph is derived, as shown in Fig. 2.

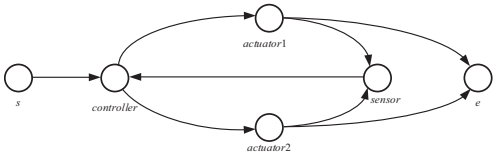

Fig. 2. Structural diagram of the dual-redundancy negative feedback control system.

When one of the actuator units fails, the node corresponding to *actuator*1 and all its associated interactions are removed, resulting in the system's structural graph for this state, as illustrated in Fig. 3.

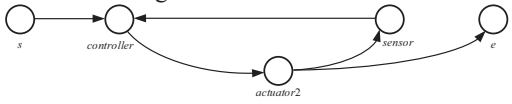

Fig. 3. Structural diagram of the dual-redundancy negative feedback control system, when one of the actuator units fails.

In addition, for certain real-world systems, the functional logic relationships between components cannot be adequately captured using the above methods. For instance, some systems incorporate "k-out-of-n" configurations, where the functional relationships among these k components cannot be described solely from a process-oriented perspective. Consequently, simplified modeling approaches are required to represent such structural configurations effectively.

Specifically, for a "k-out-of-n" link, a pair of nodes c and $c(n,k)$ defined, representing redundant component nodes connected in parallel within the graph. The structural diagram of this system is shown in Fig. 4.

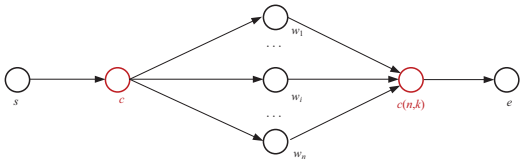

Fig. 4. Structural diagram of the "*k*-out-of-*n*" link.

The total number of paths between them, denoted as (all path, AP), is calculated as follows:

$$AP\left(c,c\left(n,k\right)\right)=n-k+1 \tag{7}$$

when there is no redundancy, n=k, and the number of paths is 1.

3.2. *System structural fault tolerance index*

Based on the system's structural graph, the redundancy of functional paths is utilized as the index to quantify the system's redundancy level. This approach is applied to analyse and discuss both SISO and MIMO systems, establishing the quantitative index to evaluate the degree of structural redundancy.

For the system structural graph $G_{\boldsymbol{\delta}}(V,E)$, the primary concern is whether the system can complete the processing from input to output. This process translates to the existence of a valid path from the start node s to the end node e in the graph.

However, not all paths from s to e can fulfil the system's intended functionality. In the structural graph, even though an open-loop path from s to e may exist without incorporating feedback from the sensor, the absence of a critical edge representing the controller's input renders the system incapable of functioning normally along such a path. Therefore, it is necessary to identify all paths from s to e and focus on those that meet the system's functional requirements. These paths are referred to as Function Paths (FP). The total number of all unique function paths is referred to as All Function Paths (AFP), and its total count is denoted as $AFP(G,s,e)$.

For the Single-Input Single-Output (SISO) system $G(V,E)$, and $V=\{s,\mathbf{c}_1,\mathbf{c}_2...,\mathbf{c}_{n_c},e\}$. When $AFP\left(G_{\boldsymbol{\delta}_0},s,e\right)>1$, The structural fault tolerance index can be expressed as:

$$\nu_s(\boldsymbol{\delta})=\frac{AFP\left(G_{\boldsymbol{\delta}},s,e\right)-1}{AFP\left(G_{\boldsymbol{\delta}_0},s,e\right)-1} \tag{8}$$

when $AFP\left(G_{\boldsymbol{\delta}_0},s,e\right)=1$ or $AFP(G_{\boldsymbol{\delta}},s,e)=0$, the system structural fault tolerance index is 0.

For the Multi-Input Multi-Output (MIMO) system $G(V,E)$, $V=\{s,\mathbf{c}_1,\mathbf{c}_2...,\mathbf{c}_{n_c},e\}$, $\mathbf{s}=\{s_1,s_2,...,s_m\}$, and $\mathbf{e}=\{e_1,e_2,...,e_n\}$. The $m\times n$ matrix $P=[p_{ij}]_{m\times n}$ describes the structural fault tolerance between various start and end points, with each element p_{ij} satisfying the following conditions:

$$p_{ij}=\frac{AFP(G_{\boldsymbol{\delta}},s_i,e_j)-1}{AFP(G_{\boldsymbol{\delta}_0},s_i,e_j)-1} \tag{9}$$

when $AFP(G_{\boldsymbol{\delta}_0},s_i,e_j)=1$ or $AFP(G_{\hat{\boldsymbol{\delta}}},s_i,e_j)=0$, the system structural fault tolerance index is 0.

Unlike SISO systems, MIMO systems exhibit two distinct scenarios: (1) a node pair is valid but lacks a functional path, leading to system failure; and (2) a node pair is invalid, and the absence of a functional path does not affect system functionality. To address this, the matrix $A=[a_{ij}]_{m\times n}$ can be introduced as a filtering mechanism. The element $a_{ij}=1$ when the node pair (s_i,e_j) serves as the start and end points of a functional path required for system operation. Otherwise, $a_{ij}=0$, representing an invalid node pair.

Define an operator $\odot$ such that the operation $U=A\odot P=[u_{ij}]_{m\times n}$ satisfies the following relationship.

$$u_{ij}=\begin{cases}p_{ij}, & \text{if} \quad a_{ij}=1\\ \varnothing, & \text{if} \quad a_{ij}=0\end{cases} \tag{10}$$

Through this process, invalid and valid start-end node pairs can be distinguished, enabling accurate quantification of the system's overall structural fault tolerance. Subsequently, the structural fault tolerance of the MIMO system is defined as the average structural fault tolerance across all valid node pairs:

$$\nu_s=\begin{cases}\dfrac{\sum_{i=1}^{m}\sum_{j=1}^{n}u_{ij}}{\sum_{i=1}^{m}\sum_{j=1}^{n}a_{ij}}, & \forall(i,j)\in\{1,2,...m\}\times\{1,2,...,n\},u_{ij}>0\\ 0, & \exists(i,j)\in\{1,2,...m\}\times\{1,2,...,n\},\ u_{ij}=0\end{cases} \tag{11}$$

Similar to SISO systems, the structural fault tolerance of the MIMO system is a non-negative value and depends on the system's state $\boldsymbol{\delta}$.

3.3. *Margin of the structural fault tolerance*

From the system safety perspective, fault tolerance must meet specified requirements. Therefore, it is necessary to define a threshold $\nu_{s\text{th}}$ for the system structural fault tolerance index to determine whether the requirements are satisfied.

Typically, systems are required to maintain at least k_f redundancy (corresponding to the AFP in the system's structural graph). Thus, the structural fault tolerance index corresponding to this redundancy requirement can be directly calculated and used as the threshold $\nu_{s\text{th}}$.

For the SISO system, there are the following relationship.

$$\nu_{s\text{th}} = \begin{cases} \dfrac{k_f - 1}{AFP(G_{\boldsymbol{\delta}_0}, s, e) - 1}, & AFP(G_{\boldsymbol{\delta}_0}, s, e) > 1 \\ 0, & AFP(G_{\boldsymbol{\delta}_0}, s, e) = 1 \text{ or } k_f = 0 \end{cases} \tag{12}$$

For the MIMO system, the requirements are the matrix $K_f = [k_{f,ij}]_{m\times n}$, and $P_{\text{th}} = [p_{\text{th},ij}]_{m\times n}$, which has the following relationship:

$$p_{\text{th},ij} = \begin{cases} \dfrac{k_{f,ij} - 1}{AFP(G_{\boldsymbol{\delta}_0}, s_i, e_j) - 1}, & AFP(G_{\boldsymbol{\delta}_0}, s_i, e_j) > 1 \\ 0, & AFP(G_{\boldsymbol{\delta}_0}, s_i, e_j) = 1 \text{ or } k_{f,ij} = 0 \end{cases} \tag{13}$$

then, by averaging the structural fault tolerance thresholds across all valid node pairs, the overall structural fault tolerance threshold for the MIMO system can be obtained:

$$\nu_{s\text{th}} = \begin{cases} \dfrac{\sum_{i=1}^{m}\sum_{j=1}^{n} u_{\text{th},ij}}{\sum_{i=1}^{m}\sum_{j=1}^{n} a_{ij}}, \forall (i,j) \in \{1,2,...m\} \times \{1,2,...,n\}, u_{th,ij} > 0 \\ 0, \exists (i,j) \in \{1,2,...m\} \times \{1,2,...,n\}, \ u_{th,ij} = 0 \end{cases} \tag{14}$$

based on this, the margin of structural fault tolerance M_{sv} can be calculated as follows.

$$M_{sv} = \nu_s - \nu_{s\text{th}} \tag{15}$$

When M_{sv} is less than or equal to 0, it means that the structural fault tolerance capability cannot meet the requirements.

4. Case Study

This section uses a typical SISO *k*-out-of-*n* system from the real-world application as a case to quantify its structural fault tolerance and margin, demonstrating the feasibility of the proposed method.

4.1. *Redundant flywheel system*

Specifically, the Redundant Flywheel System (RFS) is a commonly used attitude control system for spacecraft, designed to ensure stability in space through a redundant configuration of multiple flywheels (Firsov 2014). The system requires at least three flywheels in different directions to maintain normal functionality. If one flywheel fails, the redundant flywheels can immediately compensate, preventing attitude loss. From the functional logic perspective, the RFS is a typical *k*-out-of-*n* system with *k*= 3.

4.2. *System structural diagram of the RFS*

The input of the RFS is the desired attitude angle, while the output is the actual attitude of the inertial body. Based on its functional principles, the initial structural graph of the RFS is shown in Fig. 5. The system consists of several key components: a PD controller, a torque allocator, sensors, an inertial load, and six flywheel units.

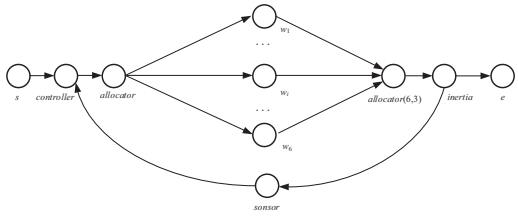

Fig. 5. Structural diagram of the RFS.

In this system structure, the *k*-out-of-*n* process occurs at the flywheel components following the torque allocator. Therefore, the pair of nodes associated with the allocator can be defined as *allocator* and *allocator*(*n*,3). The number of paths between these two nodes follows the calculation given by (7).

4.3. *System state of the RFS*

In this case, for any given flywheel unit *i*, the state δ_i of the flywheel component can be defined as:

$$\delta_i = \begin{cases} 0 & \text{Normal} \\ 1 & \text{Failure} \end{cases} \tag{16}$$

for the system state, this case study considers scenarios where the RFS has redundancy. The system state is represented by:

$$\boldsymbol{\delta} = [\delta_1, \delta_2, \delta_3, \delta_4, \delta_5, \delta_6] \tag{17}$$

let $\{\boldsymbol{\delta}_0, \boldsymbol{\delta}_1, \boldsymbol{\delta}_2, \boldsymbol{\delta}_3\}$ represent the state space of the system, where $\boldsymbol{\delta}_0 = [0,0,0,0,0,0]$, $\boldsymbol{\delta}_1 = [1,0,0,0,0,0]$, $\boldsymbol{\delta}_2 = [1,1,0,0,0,0]$, and

$\boldsymbol{\delta}_3 = [1,1,1,0,0,0]$. Under this setup, the system has a unique state transition sequence, denoted as $[\boldsymbol{\delta}_0, \boldsymbol{\delta}_1, \boldsymbol{\delta}_2, \boldsymbol{\delta}_3]$.

4.4. *Fault tolerance margin of the RFS*

Based on the above analysis and modelling, we calculate the structural fault tolerance index of the RFS by (8).

$$\nu_s(\vec{t}_j) = \frac{3-j}{3} \tag{18}$$

Furthermore, the fault tolerance requirement of the RFS necessitates at least two redundant flywheels. Substituting $k_f = 2$ into (12), we can obtain:

$$\nu_{sth} = \frac{2-1}{3} = \frac{1}{3} \tag{19}$$

As the number of flywheel failures increases $\{\boldsymbol{\delta}_0, \boldsymbol{\delta}_1, \boldsymbol{\delta}_2, \boldsymbol{\delta}_3\}$, the corresponding component failure time is denoted as $\{\vec{t}_0, \vec{t}_1, \vec{t}_2, \vec{t}_3\}$. The structural fault tolerance index and the corresponding fault tolerance margin of the system in each state can be calculated.

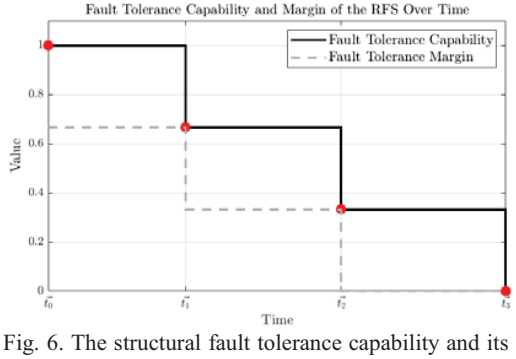

Fig. 6. The structural fault tolerance capability and its margin over time of the RFS.

The results in Fig. 6 indicate that at time $\vec{t}_2$, the margin of RFS drops to zero, meaning it can no longer meet the specified fault tolerance requirements. At this point, maintenance and contingency measures must be implemented to ensure the reliable execution of the RFS mission.

5. Discussion

The proposed method for quantifying structural fault tolerance capability offers a novel perspective that bridges component-level redundancy to system-level functionality. First, we establish the foundation for quantifying and evaluating fault tolerance through formal modeling, systematically describing the functional logic and state transitions of systems. The structured modeling approach based on directed graphs, combined with the AFP metric, effectively captures changes in functional redundancy under fault conditions.

Additionally, in terms of system design and maintenance, the proposed method holds significant practical implications. By quantifying the structural fault tolerance index and its margin, engineers can more intuitively assess a system's redundancy under fault conditions and identify critical degradation points. For instance, in the case of the RFS, the dynamic changes in the fault tolerance margin provide clear guidance for maintenance decisions, supporting a shift from reactive to proactive maintenance. Furthermore, it also can be applied to optimize redundancy design, helping designers strike a balance between cost and reliability, particularly in safety-critical domains such as aerospace and industrial control, where it has broad application potential.

6. Conclusion

This paper presents a topology-driven methodology to quantify structural fault tolerance in complex systems. By translating functional logic into directed graphs, we introduce the AFP to evaluate the degree of redundancy through valid input-to-output paths. The structural fault tolerance index normalizes redundancy under fault conditions, while threshold calculations and margin analysis provide actionable criteria for meeting the safety requirements. Unlike failure-centric approaches, the proposed method emphasizes functional sustainability, enabling designers to prioritize critical redundancies.

The case study on a redundant flywheel system validates the framework's practicality. By modeling the system as a *k*-out-of-*n* configuration, we demonstrate how AFP reductions correlate with component failures. Further, we calculate the margin of the fault tolerance and present its trajectory over the degradation time, which can support the related proactive maintenance.

In future work, we will introduce performance-oriented fault tolerance metrics to address gradual functional degradation and

integrate these with the proposed method in this paper, forming a more comprehensive and multi-scale evaluation framework.

Acknowledgement

This work is supported by the National Natural Science Foundation of China [grant number 51775020]; the National Natural Science Foundation of China [grant numbers 62073009].

References

Al-Kuwaiti, M., N. Kyriakopoulos, and S. Hussein. (2009). A comparative analysis of network dependability, fault-tolerance, reliability, security, and survivability. *IEEE Communications Surveys & Tutorials* 11 (2): 106-124.

AMARI, GREGORY LEVITIN SUPRASAD V. (2007). Reliability Analysis of Fault Tolerant Systems with Multi-Fault Coverage. *Int J Performability Eng* 3 (4): 441-451.

Bennett, J. W., G. J. Atkinson, B. C. Mecrow, and D. J. Atkinson. (2012). Fault-Tolerant Design Considerations and Control Strategies for Aerospace Drives. *IEEE Transactions on Industrial Electronics* 59 (5): 2049-2058.

Bistouni, Fathollah, and Mohsen Jahanshahi. (2014). Analyzing the reliability of shuffle-exchange networks using reliability block diagrams. *Reliability Engineering & System Safety* 132: 97-106.

Bohacek, S., J. Hespanha, J. Lee, C. Lim, and K. Obraczka. (2007). Game Theoretic Stochastic Routing for Fault Tolerance and Security in Computer Networks. *IEEE Transactions on Parallel and Distributed Systems* 18 (9): 1227-1240.

Campelo, J. C., F. Rodríguez, A. Rubio, R. Ors, P. J. Gil, L. Lemus, J. V. Busquets, J. Albaladejo, and J. J. Serrano. (1999). Distributed industrial control systems: a fault-tolerant architecture. *Microprocessors and Microsystems* 23 (2): 103-112.

Chen, Peter M., Edward K. Lee, Garth A. Gibson, Randy H. Katz, and David A. Patterson. (1994). RAID: high-performance, reliable secondary storage. *ACM Comput. Surv.* 26 (2): 145–185.

Dugan, J. B., S. J. Bavuso, and M. A. Boyd. (1992). Dynamic fault-tree models for fault-tolerant computer systems. *IEEE Transactions on Reliability* 41 (3): 363-377.

Dugan, J. B., and M. R. Lyu. (1994). System reliability analysis of an N-version programming application. *IEEE Transactions on Reliability* 43 (4): 513-519.

Firsov, S. N. (2014). Formation of fault-tolerant flywheel engine units of satellite stabilization and attitude control systems. *Journal of Computer and Systems Sciences International* 53 (4): 601-609.

Gong, M., X. Li, and L. Zhang. (2019). Analytical Inverse Kinematics and Self-Motion Application for 7-DOF Redundant Manipulator. *IEEE Access* 7: 18662-18674.

Jin, J., C. Zhu, X. Li, X. Zhang, and G. Liu. 2022. "Reliability Design and Analysis of SAR Platform Load Power Supply." 2022 3rd China International SAR Symposium (CISS), 2-4 Nov. 2022.

Krishna, and Shin. (1986). On Scheduling Tasks with a Quick Recovery from Failure. *IEEE Transactions on Computers* C-35 (5): 448-455.

Lei, Yafei, and Jixiang Meng. (2020). Structure Fault-tolerance of Arrangement Graphs. *Applied Mathematics and Computation* 381: 125287.

Li, Xiao-Yang, Yue Liu, Yan-Hui Lin, Liang-Hua Xiao, Enrico Zio, and Rui Kang. (2021). A generalized petri net-based modeling framework for service reliability evaluation and management of cloud data centers. *Reliability Engineering & System Safety* 207: 107381.

Manzone, A., A. Pincetti, and D. De Costantini. 2001. "Fault tolerant automotive systems: an overview." Proceedings Seventh International On-Line Testing Workshop, 9-11 July 2001.

Ning, Meng, Lei Shao, FangJian Chen, Mingxing Li, Chunyu Zhang, and Qiuju Zhang. (2019). Modeling and Analysis of a Modular Multilegged Robot with Improved Fault Tolerance and Environmental Adaptability. *Mathematical Problems in Engineering* 2019 (1): 8261617.