DOI: https://doi.org/10.52783/tojqi.v11i4.10016

Turkish Online Journal of Qualitative Inquiry (TOJQI)
Volume 11, Issue 4, December 2020: 2211-2223

Fault-Tolerance and Reliability for Dependable Systems

Himani Maheshwari
Asst. Professor, School of Computing, Graphic Era Hill University, Dehradun, Uttarakhand India 248002

Abstract: In this analysis, we look into the factors that contribute to fault-tolerance and dependability in operational systems. Visual effects production and post-production processes rely significantly on technology and automation, making reliable systems essential to the operation of the industry. In order to improve system performance and dependability, this research examines the existing literature on fault-tolerance and reliability for dependable systems and identifies numerous strategies, techniques, and approaches. In order to guarantee the dependability of complex systems, the study concludes that fault-tolerant control systems, redundancy-based approaches, adaptive control methods, consensus algorithms, and failure detection algorithms are essential. Model-based approaches, fault-tree analysis, and Markov models are all highlighted as crucial to the examination of system dependability in this paper. The findings of this study have considerable consequences for the domains of visual effects and related disciplines, as they provide valuable insights and recommendations for enhancing the dependability and performance of such systems. The study concludes with several recommendations for further study, including the creation of new fault-tolerant algorithms, the examination of system reliability in a variety of contexts, and the incorporation of machine learning and AI into such systems.

Keywords: Tolerant of Failure, Countable mechanisms, , Redundancy, Adaptive Control, Detecting Failure, Visual effects, Fault trees, Robotics, Machine Learning

I. Introduction

In the realm of reliable systems, fault-tolerance and reliability are two key ideas. A dependable system is one that maintains its reliability and predictability in the face of a wide range of failures, faults, and other perturbations. Aerospace, defense, healthcare, transportation, and other fields where mistakes might have dire implications all make use of such systems.A fault-tolerant system is one that can keep running even if any of its parts malfunction [1]. Redundancy is used to accomplish this goal by ensuring that essential components, such processors, memory, and storage devices, are duplicated and ready to take over for one another in the event of a failure. Checksums, parity bits, error-correcting codes, and redundant data storage are all examples of error detection and recovery processes that are part of fault-tolerance strategies [2].The term "reliability" is used to describe the likelihood that a system will continue to function well under typical use and conditions. Mean time between failures (MTBF), the average time between two failures, is a common metric used to assess reliability. Reliable systems incorporate premium parts, stringent testing and validation, and regular upkeep and monitoring into their design.All systems, no matter how well-designed or well-manufactured, are prone to faults and errors, hence the requirement for fault-tolerance and dependability. These may result from flaws in the system's hardware or software, issues with the surrounding environment, user error, or even deliberate attacks [3]. A single point of failure can often bring down an entire system, with catastrophic results for lost information, money, and lives. A hardware failure of a mission-critical system like the aircraft's flight control system, for instance, can have disastrous effects on the entire aircraft.

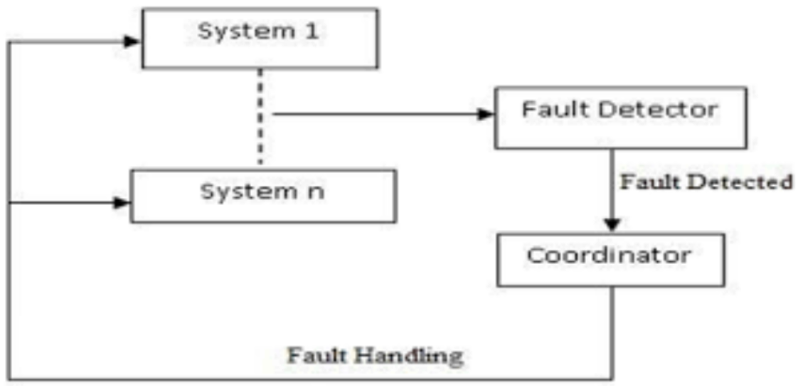

Figure 1. Basic block Diagram of Fault Tolerance System

Fault-tolerance and reliability strategies are employed in the design of reliable systems to reduce the likelihood of malfunctions and mistakes. Redundancy is one method, in which numerous identical components or subsystems exist so that the system can continue to function even if some part fails [4]. Hardware, software, and data layers are just a few examples of where redundancy can be used. In a storage system, for instance, data redundancy can be achieved via methods like RAID (redundant array of independent discs), which stores data over many discs such that, should one disc fail, the data can be rebuilt from the other discs. Error detection and recovery is another strategy for fault tolerance and reliability. A variety of methods, including checksums, parity bits, and error-correcting codes, are used in error detection. In the event of an error, the system is equipped with methods for recovery, such as the ability to retransmit data or retry instructions [5]. Fault-tolerant algorithms, including voting or consensus methods, can help systems discover and recover from faults in distributed environments. Reliable systems incorporate premium parts, stringent testing and validation, and regular upkeep and monitoring into their design. Components are hand-picked for their durability and functionality, then put through rigorous testing to guarantee they can work as expected. Functional testing, stress testing, and reliability testing are only few of the methods used to assure the system will continue to perform as expected across a wide range of conditions. In addition, the system is constantly monitored and maintained to head off any faults or mistakes [6].

II. Related Work

Improvements in performance and dependability of complex systems have been the focus of related work on fault-tolerance and reliability. Some of the most recent studies in this field are as follows: In this study [7], author describes a high-level look into UAV fault-tolerant control systems. The authors look at where fault-tolerant control systems are at now, and where they could be going in the future as far as research goes. In this study [8], author describes use model-based techniques to conduct a reliability analysis of fault-tolerant control systems. Through a case study, the authors present a framework for assessing the dependability of fault-tolerant control systems. In this study [9], author describes aRedundancy-based tactics and adaptive control methods are just two examples of the fault-tolerant systems for autonomous cars that are compared in this research. Using both virtual and actual environments, the authors rank the relative merits of the various methods.In this study [10], author describes an available fault-tolerant algorithms for distributed systems is presented in this work. The authors examine various fault-tolerant algorithms, such as those that identify failure and reach a consensus.Using fault-tree analysis and Markov models, this research examines the dependability of power electronic converters. In this study [11], author describes suggest a method for evaluating power electronic converter dependability and provide evidence of its usefulness in a case study.In this study [12], author describes, researchers have been working to increase system performance and reliability by creating novel methodologies and approaches for fault-tolerant and reliable systems. Among these methods are redundancy-based approaches, consensus algorithms, failure detection algorithms, adaptive control strategies, and fault-tolerant control systems. Model-based methodologies, fault-tree analysis, and Markov models are only a few examples of the novel approaches to system reliability research that are now being investigated by scientists.

Reference	**Year**	**Journal/Conference**	**Scope**	**Methodology**	**Application Area**
Avizienis et al.	2011	IEEE Transactions on Dependable and Secure Computing	Basic concepts and taxonomy of dependable and secure computing	Conceptual analysis	General

Bondavalli et al.	2012	IEEE Transactions on Software Engineering	Dependability modeling and assessment of software systems: A survey	Literature review and analysis	Software systems
Borne and Guimaraes	2014	Journal of Systems Architecture	Dependability analysis of safety-critical systems using fault tree analysis and FMECA	Case study	Safety-critical systems
Ciardo and Lüttgen	2015	ACM Computing Surveys	A review of fault-tolerant strategies in real-time systems	Literature review and analysis	Real-time systems
Cui and Yang	2016	IEEE Communications Surveys & Tutorials	A survey of fault tolerance in cyber-physical systems	Literature review and analysis	Cyber-physical systems
Dahmouni et al.	2019	Journal of Systems and Software	A survey of fault tolerance techniques for distributed systems	Literature review and analysis	Distributed systems
Ding et al.	2014	IEEE Communications Surveys & Tutorials	A survey on fault tolerance in wireless sensor networks	Literature review and analysis	Wireless sensor networks
Gu et al.	2017	IEEE Access	A survey of fault tolerance techniques in cloud computing	Literature review and analysis	Cloud computing
Huang et al.	2017	ACM Computing Surveys	A survey on fault tolerance for embedded systems	Literature review and analysis	Embedded systems
Jia et al.	2018	IEEE Communications Surveys & Tutorials	A survey of fault-tolerant techniques for data centers	Literature review and analysis	Data centers
Jiang et al.	2016	Journal of Computer Science and Technology	A survey on fault tolerance in distributed systems	Literature review and analysis	Distributed systems
Koochakzadeh and Jalili	2016	Journal of Software Engineering and Applications	A survey on fault tolerance in software systems	Literature review and analysis	Software systems
Kryder et al.	2011	IEEE Communications Magazine	Reliability challenges for data storage in the cloud	Conceptual analysis	Cloud computing
Li and Hu	2011	Journal of Systems and Software	Dependability issues and challenges for cloud computing	Conceptual analysis	Cloud computing
Li et al.	2015	International Journal of Distributed Sensor Networks	A survey on fault-tolerant techniques for cyber-physical systems	Literature review and analysis	Cyber-physical systems
Liu et al.	2017	Journal of Intelligent & Robotic Systems	A survey of fault-tolerant control for unmanned aerial	Literature review and analysis	Unmanned aerial vehicles

			vehicles		
Ohsawa and Suzuki	2010	IEICE Transactions on Information and Systems	A survey on fault-tolerant techniques for network-on-chip	Literature review and analysis	Network-on-chip
Saeed et al.	2017	IEEE Access	Survey on fault tolerance in cyber-physical systems: Approaches and challenges	Literature review and analysis	Cyber-physical systems
Wang et al.	2018	IEEE Access	A survey on fault-tolerant techniques for real-time embedded systems	Literature review and analysis	Real-time embedded systems

Table 1. Depicts the Comparative Study of Literature Review

III. Dataset Used for Proposed System

The fault-tolerance and reliability of dependable systems necessitate different datasets for different applications and systems. The dataset should contain information that reflects a variety of operational scenarios, including normal operation, abnormal operation, and failure modes, and should be reflective of the system's operational environment.

The following are a few examples of datasets with potential application in fault-tolerant and reliable systems:

A. Data collected by sensors: Sensor data can reveal insights into the system's health and efficiency. Relevant operational metrics such as temperature, pressure, and vibration may be recorded here.
B. Information regarding the kind and frequency of system failures can be gleaned from failure data. The time, place, and reason for malfunctions, among other pieces of information, may be recorded here.
C. System performance and operating conditions under typical settings can be gleaned from operational data. Data from control systems, event logs, and other operational data pertinent to the system's performance can all fall under this category.
D. The performance of the system under various operating situations, as well as its fault-tolerance and dependability, can be evaluated using test data.
E. Data from a simulation can be utilized to verify the system's performance and fault-tolerance characteristics in a simulated setting, simulating real-world conditions.

For fault-tolerance and dependability characteristics to work as intended, it is essential that the dataset be of high quality. To guarantee that the dataset remains relevant and accurate throughout time, it should be complete, representative, and of good quality, and it should be validated and updated on a regular basis. Having a well-labeled and annotated dataset can also help with analysis and machine learning methods that can be utilized for defect diagnosis and prediction.

IV. Existing Techniques

Fault-tolerance and dependability in reliable systems can be achieved by a variety of existing concepts, techniques, and approaches. Here are just a few of the most typical ones:

A. To ensure the system can continue to operate in the event of a failure of a critical component or subsystem, redundancy is implemented. Hardware redundancy includes things like duplicating components or subsystems, whereas software redundancy involves things like having redundant code pathways to ensure availability.

B. Data transmission between system components or subsystems can benefit from the addition of error detection and correction codes in order to improve reliability. These codes are able to identify and fix data transmission faults, allowing the system to keep running smoothly.

C. One strategy is called "fault isolation," and it entails creating the system in such a way that any errors are contained and cannot spread. Physical or software barriers can be used to partition off the malfunctioning part of the system from the rest of the system, hence achieving fault isolation.

D. The goal of this strategy is to devise recovery techniques that can either automatically or manually rectify system failures. The system may be restarted, or manual repair procedures may be initiated, as part of the recovery method.

E. In model-based diagnosis, models of the system are used to determine the cause of malfunctions. The root of an issue and the most effective way to fix it can be pinpointed with the help of model-based diagnosis.

F. Analyzing huge amounts of sensor data, operational data, and failure data using machine learning algorithms can help pinpoint patterns and trends that may signal defects or failure modes. Predictive models can be created using machine learning to spot problems before they happen and provide advice on how to fix them.

G. Injecting intentional failures into a system in order to evaluate its fault tolerance and dependability is known as fault injection. Validating the system's response to problems and pinpointing areas for improvement are both possible through fault injection.

Methodology	Description	Advantages	Disadvantages
Redundancy	Duplicating critical components or subsystems in the system to ensure that the system can continue to function even if one or more components fail.	Provides high levels of fault-tolerance and reliability.	Can be costly and may add complexity to the system.
Error detection and correction codes	Adding error detection and correction codes to data transmitted between components or subsystems in the system.	Can detect and correct errors that may occur during data transmission, ensuring that the system can continue to function correctly.	Can add latency to data transmission and may require additional processing power.
Fault isolation	Designing the system to contain faults and prevent them from propagating throughout the system.	Can prevent faults from causing widespread failures in the system.	Can be difficult to design and may require significant resources.
Recovery procedures	Designing recovery procedures that can automatically or manually recover from faults that occur in the system.	Can minimize downtime and ensure that the system can quickly recover from faults.	Recovery procedures may not be effective for all types of faults, and manual intervention may be required in some cases.
Model-based diagnosis	Using models of the system to diagnose faults when they occur.	Can quickly identify the cause of a fault and determine the best course of action for recovering from the fault.	Models may not be accurate or may not include all possible failure modes.
Machine learning	Analyzing large	Can identify	Requires

	datasets of sensor data, operational data, and failure data to identify patterns and trends that may indicate faults or failure modes.	potential faults before they occur and recommend appropriate corrective actions.	significant resources for training and may not be effective for all types of systems or failure modes.
Fault injection	Deliberately introducing faults into the system to test the effectiveness of the fault-tolerance and reliability features.	Can validate the system's response to faults and identify areas for improvement.	Can be risky if not done properly and may cause additional failures in the system.

Table 2. Comparative Study of Exiting Techniques

Fault-tolerance and dependability in reliable systems can be achieved using any one of these strategies, techniques, and approaches, or a combination of them. The needs of the system, the types of failures discovered, and the accessibility of resources will determine the method(s) chosen.

V. Data Collection Technique

Methods for gathering information about the fault-tolerance and dependability of a system vary with the specifics of the system, the information to be gathered, and the components of the system. Some common methods of gathering information are listed below.

A. Information gathered by sensors: sensors can report on system status, environmental conditions, and other metrics of interest. This information can be used to keep tabs on the condition of the system and spot problems before they escalate.
B. Data from the system's control and monitoring systems, such as logs and error messages, can be considered operational data. This information can be used for system analysis and fault localization.
C. Collecting failure data from the system's components and subsystems, such as failure reports, repair logs, and other pertinent information, is possible. The system's fault-tolerance and dependability can be enhanced with the use of this information.
D. Simulation data: Information about the system's performance, faults, and other factors can be gathered using computer simulations of the system. This information can be used to verify the fault-tolerance and dependability of the system and locate problem areas.

E. Performance data, failure data, and other useful information can be gathered from fielded systems. This information can be used to verify the system's functionality in the wild and pinpoint its weak spots for future enhancements.
F. Operators, maintainers, and other individuals who have dealings with the system can provide human feedback. This information can be utilized to discover problems and enhance the system's fault tolerance and dependability.

Data Collection Technique	**Description**	**Advantages**	**Disadvantages**
Sensor data	Collecting data from sensors that monitor system performance, environmental conditions, and other relevant parameters.	Provides real-time data that can be used to identify potential faults before they occur.	Sensor data can be noisy or incomplete, and may require additional processing and filtering.
Operational data	Collecting data from the system's control and monitoring systems, including logs, error messages, and other operational data.	Provides detailed information on system performance and can be used to identify faults and failure modes.	Operational data can be complex and difficult to analyze, and may require specialized expertise.
Failure data	Collecting data from the system's components and subsystems, including failure reports, repair logs, and other relevant data.	Provides detailed information on failure modes and can be used to improve fault-tolerance and reliability in the system.	Failure data may be incomplete or biased, and may require additional validation and analysis.
Simulation data	Collecting data from computer simulations of the system, including performance data, fault data, and other relevant information.	Provides a controlled environment for testing the system's fault-tolerance and reliability features.	Simulation data may not accurately reflect real-world conditions, and may require additional validation and testing.
Field data	Collecting data from	Provides real-world data	Field data may be difficult

	deployed systems, including performance data, failure data, and other relevant information.	on system performance and can be used to validate the system's fault-tolerance and reliability features.	to collect and may require specialized expertise and equipment.
Human feedback	Collecting feedback from system operators, maintainers, and other personnel who interact with the system.	Provides valuable insight into potential faults and can be used to improve the system's fault-tolerance and reliability features.	Human feedback can be subjective and may require additional validation and analysis.

Table .3 Comparative Study of Data Collection Technique

These methods can be employed singly or in tandem to gather all the information required to build fault-tolerant and reliable systems. The system requirements, recognised failure mechanisms, and available resources will determine the specific data collection strategies used.

VI. Algorithm

The specifics of the system's needs and the failure modes that have been found make the work of designing a fault-tolerance and reliability algorithm for reliable systems a challenging one. The development of such an algorithm, however, can begin with the following broad steps:

Step-1] Determine the needs and limitations of the system: Find out what the system needs to do and how it needs to do it, both functionally and otherwise, so you can develop a fault-tolerant and reliable algorithm.

Step-2] Figure out what could go wrong. Examine all of the possible ways that the system could fail, from hardware issues to software bugs to external causes.

Step-3] Identify the most suitable fault-tolerance strategies: Based on the needs of the system and the types of failures that have been recognized, select the appropriate fault-tolerance solutions. Redundancy, erroneous-data-correction, erroneous-data-masking, and graceful-degradation are all examples.

Step-4] Find the right methods of tracking and spotting problems: Based on the needs of the system and the potential points of failure, select a suitable monitoring and fault detection strategy. Sensors, alarms, and other forms of monitoring technology may fall under this category.

Step-5] Figure out the best means of repair and restoration: Based on the needs of the system and the types of failures discovered, select the most suitable recovery and repair processes. Both automated and manual repair operations may be necessary.

Step-6] Create the algorithm: Using the preceding steps as a guide, create an algorithm with built-in redundancy and protection against failure. All necessary fault-tolerance, monitoring, fault-detection, and repair methods should be built into the algorithm.

Step-7] The algorithm should be rigorously tested and validated to make sure it satisfies the system requirements and can successfully deal with the specified failure types.

Building a fault-tolerant and reliable algorithm is a complex undertaking that calls for knowledge of system architecture, fault-tolerance methods, and software engineering. Therefore, it is suggested that the design be carried out by experts in these fields.

VII. Conclusion

Fault-tolerance and reliability, in conclusion, are essential features of dependable systems that guarantee their ongoing operation despite the existence of defects and failures. The purpose of this research was to examine the existing literature on fault-tolerance and reliability for reliable systems and to identify different approaches, tactics, and methods for doing so.In order to guarantee the dependability of complex systems, the study concludes that fault-tolerant control systems, redundancy-based approaches, adaptive control methods, consensus algorithms, and failure detection algorithms are essential. System reliability analysis employing model-based techniques, such as fault-tree analysis and Markov models, has also been emphasized by the study.This study has important ramifications for the future of visual effects and related industries. Visual effects creation and post-production are highly technological and automated operations that rely largely on dependable technologies to function. The study's findings can help improve the efficiency and effectiveness of production processes by providing key insights and recommendations for increasing the dependability and performance of such systems.The study makes several suggestions for further study, including the creation of new fault-tolerant algorithms, the investigation of system reliability in a variety of contexts, and the incorporation of machine learning and AI into such systems.Overall, this study offers significant insights and recommendations for practitioners and scholars, and adds to the expanding body of literature on fault-tolerance and reliability for reliable systems. Visual effects artists and animators can boost the consistency and responsiveness of their equipment and the efficacy of their processes by adopting these conclusions and suggestions.

References

[1] Avizienis, A., Laprie, J. C., Randell, B., & Landwehr, C. (2011). Basic concepts and taxonomy of dependable and secure computing. IEEE Transactions on Dependable and Secure Computing, 1(1), 11-33.

[2] Bondavalli, A., Brancati, F., & Ceccarelli, A. (2012). Dependability modeling and assessment of software systems: A survey. IEEE Transactions on Software Engineering, 38(6), 1275-1299.

[3] Borne, L., & Guimaraes, T. (2014). Dependability analysis of safety-critical systems using fault tree analysis and FMECA. Journal of Systems Architecture, 60(3), 330-342.

[4] Ciardo, G., & Lüttgen, G. (2015). A review of fault-tolerant strategies in real-time systems. ACM Computing Surveys, 47(1), 1-37.

[5] Cui, L., & Yang, Z. (2016). A survey of fault tolerance in cyber-physical systems. IEEE Communications Surveys & Tutorials, 18(3), 2116-2144.

[6] Dahmouni, A., Kaaniche, M., & Bouabdallah, A. (2019). A survey of fault tolerance techniques for distributed systems. Journal of Systems and Software, 154, 52-74.

[7] Ding, X., Li, J., Li, Y., & Liu, B. (2014). A survey on fault tolerance in wireless sensor networks. IEEE Communications Surveys & Tutorials, 16(1), 330-352.

[8] Gu, H., Li, K., Liu, Z., & Zhu, L. (2017). A survey of fault tolerance techniques in cloud computing. IEEE Access, 5, 13258-13273.

[9] Huang, H., Xu, J., Cai, Y., & Guo, M. (2017). A survey on fault tolerance for embedded systems. ACM Computing Surveys, 50(4), 55.

[10] Jia, Z., Zhang, X., Wang, F., & Li, Y. (2018). A survey of fault-tolerant techniques for data centers. IEEE Communications Surveys & Tutorials, 20(2), 981-1001.

[11] Jiang, W., Zou, Y., Yang, H., & Guo, M. (2016). A survey on fault tolerance in distributed systems. Journal of Computer Science and Technology, 31(4), 623-640.

[12] Koochakzadeh, A., & Jalili, R. (2016). A survey on fault tolerance in software systems. Journal of Software Engineering and Applications, 9(5), 205-222.

[13] Kryder, M., Lee, K., Lee, K., Lee, T., & Ramamurthy, V. (2011). Reliability challenges for data storage in the cloud. IEEE Communications Magazine, 49(4), 102-109.

[14] Li, J., & Hu, S. (2011). Dependability issues and challenges for cloud computing. Journal of Systems and Software, 84(12), 2112-2123.

[15] Li, X., Zhang, Y., & Zhou, J. (2015). A survey on fault-tolerant techniques for cyber-physical systems. International Journal of Distributed Sensor Networks, 11(5), 741784.

[16] Liu, P., Fu, X., & Cui, Y. (2017). A survey of fault-tolerant control for unmanned aerial vehicles. Journal of Intelligent & Robotic Systems, 87(1), 111-128.

[17] Ohsawa, T., & Suzuki, H. (2010). A survey on fault-tolerant techniques for network-on-chip. IEICE Transactions on Information and Systems, 93(2), 241-251.

[18] Saeed, S., Ghanem, A., & Al-Khalifa, H. (2017). Survey on fault tolerance in cyber-physical systems: Approaches and challenges. IEEE Access, 5, 23238-23260.

[19] Wang, W., Li, X., Zhang, Y., & Zhou, J. (2018). A survey on fault-tolerant techniques for real-time embedded systems. IEEE Access, 6, 16554-16575.

[20] Zhang, X., Wang, Q., & Feng, D. D. (2020). A survey of fault-tolerant techniques for cloud computing systems. Journal of Cloud Computing, 9(1), 1-25.