Analysis of Design Parameters in Safety-Critical Computers

HAMZEH AHANGARI, FUNDA ATIK, YUSUF IBRAHIM OZKOK, ASIL YILDIRIM, SERDAR OGUZ ATA, AND OZCAN OZTURK (Member, IEEE)

H. Ahangari, F. Atik, and O. Ozturk are with the Department of Computer Engineering, Bilkent University, Ankara 06800, Turkey
Y.I. Ozkok, A. Yildirim, and S.O. Ata are with the Aselsan Corporation, Ankara 06172, Turkey
CORRESPONDING AUTHOR: H. AHANGARI (hamzeh@bilkent.edu.tr)

ABSTRACT Nowadays, safety-critical computers are extensively used in many civil domains like transportation including railways, avionics, and automotive. In evaluating these safety critical systems, previous studies considered different metrics, but some of safety design parameters like failure diagnostic coverage (C) or common cause failure (CCF) ratio have not been seriously taken into account. Moreover, in some cases safety has not been compared with standard safety integrity levels (IEC-61508: SIL1-SIL4) or even have not met them. Most often, it is not very clear that which part of the system is the Achilles heel and how design can be improved to reach standard safety levels. Motivated by such design ambiguities, we aim to study the effect of various design parameters on safety in some prevalent safety configurations, namely, 1oo2 and 2oo3, where 1oo1 is also used as a reference. By employing Markov modeling, we analyzed the sensitivity of safety to important parameters including: failure rate of processor, failure diagnostic coverage, CCF ratio, test and repair rates. This study aims to provide a deeper understanding on the influence of variation in design parameters over safety. Consequently, to meet appropriate safety integrity level, instead of improving some parts of a system blindly, it will be possible to make an informed decision on more relevant parameters.

INDEX TERMS Safety-critical computer, IEC 61508, random hardware failure, common cause failure, Markov modeling

I. INTRODUCTION Nowadays, safety-critical computers are obligatory constituents of many electronic systems that effect human life safety. Several areas of transportation industry like railways, avionics, and automotive, increasingly use such systems. To design a computer for safety-critical applications, industrial safety levels such as international safety standard IEC 61508 (shown in Table 1) have been set. In this domain, safe microcontrollers with limited processing capabilities are available in the market for mostly control purposes. However, as systems become more and more complex and versatile, having safe processors with intensive processing capabilities becomes an essential need. According to IEC 61508-2 standard, a single processor can achieve at most SIL3 level. In most cases, safety critical applications in civil domains require a higher level of safety, such as SIL4. Hence, to answer this eminent need, a computing platform needs to be architected in system level with safety in mind.

In order to achieve such high standards, it is necessary to make improvements in numerous aspects of a general purpose system. Reliability of electronic components is the most obvious factor that needs to be satisfied for building a robust system. Besides, clever system design by means of available electronic components is as important as the quality of components themselves. Even with reliable and robust parts, safety goals may not be achieved without safety aware design process. Prevalent design issues like perfect printed circuit board, EMC/EMI isolation, power circuitry, failure rates of equipment etc., are examples of common quality considerations. However, in critical systems, in addition to these, some other less obvious issues have to be observed.

Redundancy, meaning having multiple processors (channels) doing a same task, is required by safety standards in many cases. The ratio of Common Cause Failures (CCFs), defined as the ratio of concurrent failure rate (simultaneous failures among redundant channels) over total failure rate,
has great impact on system safety. The percentage of failures the system is able to detect by means of fault detection techniques also has a direct effect on safety. This is because undetected failures are potential dangers. To remove such undetected failures, an important factor is the frequency and the quality of system maintenance. Frequency and comprehension of the system tests (automatically or by technicians) to repair or replace the impaired components, can guarantee the required safety level by removing transient failures or refreshing worn-out parts.

As safety is a very wide subject, the main objective of this paper is to investigate the sensitivity of system safety, affected by random hardware failures, to some crucial design parameters. Three widespread configurations; 1oo1, 1oo2, and 2oo3; with known values of parameters, are assumed as base systems. For these systems, we evaluated individual parameters that contribute to safety. In this paper, we target high demand/continuous systems, where the frequency of demand to run the safety function is more than one per year, unlike low demand systems where it is less than one per year [17]. Average frequency of a dangerous failure of safety function per hour (PFH), is the safety measure for high demand/continuous systems, while probability of failure on demand (PFD), is the measure for low demand systems. PFH is defined as average rate of entering into unsafe state, while PFD is defined as probability of being in unsafe state.

This paper is organized as follows: In Section II, some of the recent or relevant works are reviewed and our motivation is given in more detail. In Section III, definition and modeling for considered design parameters are described. In Section IV, base systems and their Markov modeling are proposed. In Section V, experimental results are discussed, while Section VI discusses a simplified Markov modeling in safety calculations. Finally, conclusion is given in Section VII.

II. RELATED WORKS AND MOTIVATION

During the design process, concentrating on multiple aspects of design altogether for the purpose of improvement can be complicated. Normally, if the prototype design does not meet the requirements, it is rational to find the system’s bottleneck and focus on it. In safety-related designs, by knowing the share that each parameter provides to safety, the designer can decide where to put more emphasis to improve the outcome with least amount of effort. Here, we discuss some of the safety-critical computer system designs in literature, which considered a subset of safety design parameters due to complexity.

In [16], authors designed a redundant computer system for critical aircraft control applications, and an acceptable level of fault tolerance is claimed to be achieved with using five redundant standard processors, extensive error detection software and fault isolation mechanism. In [6], dual-duplex and Triple Modular Redundancy (TMR) synchronous (with common clock signal) computer systems have been built using military and commercial electronic parts. While authors tried to improve the system safety, the effect of CCFs is not assessed, where this effect can be significant in synchronous systems. Besides, the achieved safety level is not compared to any standard level. In microcontroller-based SIL4 software voter [9], SIL4 level is claimed to be obtained with a duplex architecture. Nevertheless, neither failure coverage, nor CCFs are assessed in sufficient details. Similarly, in [5], authors target safe computer system for a train, which is not compared to standard levels, and does not consider CCFs or diagnostic coverage.

These approaches, either lack of considering some of the most influential safety design parameters or methodology to assess the system safety level with respect to standards. Thus, these studies are incomplete to be considered for real safety-critical applications due to complexity of taking all parameters into account. This stimulated us to have an analysis on a few safety architectures also used in above studies. By showing the sensitivity of safety to each such parameter, we aim to provide a comparative understanding of these occasionally ignored parameters. This can help practitioners to select most appropriate parameter for improving the safety. Depending on the constraints, the most appropriate parameter can be translated to the one that leads to cheapest, fastest, or easiest system modification (as shown in Figure 1).

In [13], authors model a safety-related system in low demand mode using Markov chain to calculate PFD measure, in a way that is explained in the respective standard [19]. Several parameters such as CCF, imperfect proof testing, etc. are integrated into the model to investigate their influence over safety. However, in our work, we focus on PFH, where its calculation is not as straightforward as PFD. Moreover, we include additional parameters such as frequency of online testing, self-testing, etc., with sensitivity analysis for each parameter.

There have been many efforts related to generalized and simplified PFH formula for M-out-of-N (MooN) architectures. The works proposed in [4], [12] develop a set of
analytical expressions with some assumptions and parameters different than ours, like considering partial proof test, slightly taking the CCF contributions into account or dealing with dangerous detected failures differently. In [11], probabilistic analysis of safety for MooN architectures is proposed when considering different degrees of uncertainty in some safety parameters such as failure rate, CCFs, and diagnostic coverage, by combining Monte Carlo sampling and fuzzy sets. Emphasizing the significance of CCF impact over safety in redundant systems, in [3], authors explore the criticality of beta-factor on safety calculations. Specifically, they address PFD measure for a typical 1oo2 system. Influence of diversity in redundancy (i.e., implementing redundancy with components technologically diverse) over CCF is assessed in [15] by a design optimization approach for low demand systems.

III. SAFETY PARAMETERS IN OUR ANALYSIS
In this section, we review the definition and modeling of design parameters that effect safety.

A. PROCESSOR FAILURE RATE
A safety-critical computer system is composed of one or more redundant processors, connected to each other by communication links. We may also call them as channels or programmable electronics (PEs) according to the safety standards terminology. Generally, there is no extraordinary requirement regarding reliability of PEs. Due to low quantity and high cost of these systems, components are not necessarily designed for reliability purposes. Most often, a PE is a standard processor module, built from available Commercial-Off-The-Shelf (COTS) electronic parts including processor, memory, power circuitry, etc. In this study, we take a PE as a black box, assuming it comes with a single overall failure rate, $\lambda_{PE}$.

B. COMMON CAUSE FAILURE (CCF)
According to IEC 61508-4 standard [17], Common Cause Failure (CCF, or dependent failure) is defined as concurrent occurrence of hardware failures in multiple channels (PEs) caused by one or more events, leading to system failure. If it does not lead to a system failure, it is then called common cause fault. The $\beta$ factor represents the fraction of system failures that is due to CCF. Typically, for a duplicated system, $\beta$ value is around a few percent, normally less than 20 percent. In safety standards, two $\beta$ values are defined for detected and undetected failures ($\beta_d$ and $\beta$), while here we only assume a single $\beta$ value for both. Assuming that the CCF ratio between two PEs is taken as $\beta$, by using the extended modeling and notations shown in [8], we make the following observations for all systems in this work:

- 1oo1 configuration: Since there is no redundant PE, $\beta_{1oo1} = 0$.
- 1oo2 configuration: As depicted in Figure 2, the $\beta$ of system is only related to CCFs between two PEs. Therefore, $\beta_{1oo2} = \beta$.

As formulated in [7], we use the following expressions to describe the system’s $C$ rate. According to the referred formulation, the total $C$ is expressed as

$C = C_{selftest} + C_{compare} \approx 1.$

More specifically, $k$ is the efficiency of comparison test. Since the comparison method is more effective against independent failures (non-CCFs), it is reasonable to differentiate between $C$ rate of CCF and independent failures. Therefore, two variants of former expression can be derived

$C' = C_{selftest} + (1 - C_{selftest}) \cdot k.$

Here $k'$ and $k''$ are two constants, $0 \leq k', k'' \leq 1$, describing the efficiency of comparison for either of two classes of failures. Since comparison is less effective against CCFs, the $k''$ value is low, generally less than 0.4, while $k'$ can be close to

**FIGURE 2. $\beta$ models for duplicated and triplicated systems [8].**
Here, $\beta_d = 0.3$ of $\beta$, while $\beta$ is split into $0.3\beta + 0.7\beta$.

- 2oo3 configuration: As depicted in Figure 2, the overall $\beta$ of 2oo3 system is related to mutual CCFs, plus CCFs shared among all three PEs. Note that by definition, the CCF ratio between every two PEs is taken as $\beta$. The $\beta_2$ is defined as a number in $[0, 1]$ range, expressing part of $\beta$ which is shared among all three PEs [8]. For a typical 2oo3 system, we assume $\beta_2 = 0.3$ of $\beta$, making $\beta_{2oo3} = 2.4\beta$ (see Figure 2).

The two parameters, $\beta$ and $\beta_2$, indicators of mutual and trilateral PEs isolation, are evaluated in our analysis.

C. FAILURE DIAGNOSTIC COVERAGE
According to IEC 61508-4 [17], Diagnostic Coverage (C or DC) is defined as the fraction of failures detected by automatic online testing. Generally two complementary techniques are employed to detect failures, self-testing and comparison. Self-testing routines run upon each PE to diagnose occasional failures autonomously and they usually detect absolute majority of failures, normally around 90 percent. Second diagnostic technique is data comparison among redundant PEs for detecting the rest of the undetected failures. Hence, generally we can express $C$ as

$C = C_{selftest} + C_{compare} \approx 1.$
one [7]. Therefore, normally $C^c \leq C^f$. Three representative parameters: $C_{selftest}, k^c$ and $k^f$; are used in our analysis.

**D. TEST AND REPAIR**

Based on the IEC 61508 standard, two forms of test and repair have to be accessible for safety systems: online test and proof test. In online test (or automatic test), diagnostic routines run on each PE periodically, while system is available. As soon as a failure is detected, the faulty PE (or in some configurations the whole system) is supervised to go into fail-safe mode to avoid dangerous output. Thereafter, system tries to resolve the failure with immediate call for personnel intervention or a self commanded restart without human intervention. For transient failures, a system restart can be a fast solution, while for persistent failures switching to a spare PE or system, provides a faster recovery. In any case, online repairing is supposed to last from a few minutes to a few days. Repair rate is denoted by $\mu_{OT}$ which is defined as $1/MTTR_{OT}$ ($MRT$: mean repair time). The $t_D$ parameter is the time to detect a failure in online testing. There is no direct reference to this parameter in the standard, probably because it is assumed to be negligible with respect to repair time. However, it has been considered in literature [10]. $MTTR_{OT}$ (mean time to restoration) is mean total time to detect and repair a failure (see Figure 3). Some systems may support partial recovery which means repairing a faulty PE, while the whole system is operational. Restarting only the faulty PE (triggered by operational PEs), can make it operational again. However, if the fault is persistent, such recovery is not guaranteed. Three parameters $t_D, \mu_{OT}$ and availability of partial recovery are also considered in our analysis.

Proof test (or offline test or functional test) is the second and less frequent form of testing, whereby the periodic system maintenance process is performed by technicians. During such a maintenance, system is turned off and deeply examined to discover any undetected failure (not detected by online diagnostics) followed by a repair or replacement of defective parts. Test Interval ($TI$) defines the time interval at which this thorough system checking is performed and is typically from a few weeks to a few years. In such a scenario, repair time is negligible relatively. $MTTR_{PT}$ is the mean time to restoration (detect and repair as shown in Figure 3) from an undetected failure, and on average is taken as $TI/2$ [19]. Proof test and repair rate is denoted by $\mu_{PT} = 1/MTTR_{PT}$.

The $\mu_{PT}$ is another parameter considered in our analysis.

**IV. BASE SYSTEMS**

In this section, we define two prevalent safety configurations, 1oo2 and 2oo3 plus simple 1oo1 as a reference, for our analyses. Configurations are modeled by Markov chain with employing all the aforementioned parameters. The assigned set of default values for parameters specify the initial safety point for each system.

**A. ASSUMPTIONS**

In this study, we make the following assumptions: Typically, a safe computer is responsible for running user computations. At the same time, it is in charge of checking the results for possible failures and taking necessary measures (in other words, running safety function). Specifically, the safety function of the system detects and prevents any erroneous calculation result on PEs. All PEs are asynchronous and identical (homogeneous) and connected to each other by in-system links, whereby software voting and comparison mechanisms operate (Figure 4). In this work, our focus is on processors (PES), while I/O ports and communication links are assumed to be black-channel, by which safety is not affected. This assumption can be realized by obeying standards applied for safe communication over unsafe mediums (e.g., EN-50159). These systems are assumed to be single-board computers (SBCs), meaning all redundant PEs reside on one board. Online repair of a PE with detected failures makes it operational again, but a PE with undetected failures can be repaired only by proof test and repair. Moreover, for biasing a high demand/continuous system toward safety, degraded operation is not allowed. It means when a failure is detected, the faulty PE activates its fail-safe output and contributes to voting (alternatives are 1-to report the failure but keep the PE’s output silent, leading to degradation of system, for example from 1oo2 to 1oo1, or 2-to not tolerate any faulty PE [12]). A PE with undetected failure is assumed to be seemingly operational and able to run diagnostic routines. Another simplifying assumption is that a CCF occurs in a symmetric way across PEs in a way that it is detectable on all PEs or none of them.
TABLE 2. Default values for safety design parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Meaning</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\lambda_{PF}$</td>
<td>PE failure rate</td>
<td>1.0E-5/hour</td>
</tr>
<tr>
<td>$C_{selftest}$</td>
<td>Diagnostic coverage of self-testing [7]</td>
<td>0.90</td>
</tr>
<tr>
<td>$k'$</td>
<td>Comparison efficiency for independent failures [7]</td>
<td>0.90</td>
</tr>
<tr>
<td>$\beta$</td>
<td>CCF ratio between each two PEs</td>
<td>0.02</td>
</tr>
<tr>
<td>$\beta_2$</td>
<td>Part of $\beta$ shared among three PEs [8]</td>
<td>0.3</td>
</tr>
<tr>
<td>$t_D$</td>
<td>Time to detect failure by online test (inverse of online test rate)</td>
<td>0 (negligible)</td>
</tr>
<tr>
<td>$\mu_{OT}$</td>
<td>Online repair rate</td>
<td>1/hour</td>
</tr>
<tr>
<td>$\mu_{PT}$</td>
<td>Proof test and repair rate</td>
<td>0.0001/hour (≈ 1/year)</td>
</tr>
</tbody>
</table>

B. DEFAULT PARAMETERS

For safety parameters which we intend to investigate in this study, we assign a set of default values to define an initial safety point for each system (shown in Table 2). In our experiments, we sweep each parameter around the default value and illustrate how safety is affected. This way, the sensitivity of system safety with respect to that parameter will be revealed. Although in implementing a system, some parameters such as $\beta$ and $\beta_2$, or $k'$ and $k''$, may not be practically independent, but we disregard this dependency which is due to implementation.

C. MARKOV MODELS

In this section, we give our safety configurations modeled by Markov chain. Reliability, Availability, Maintainability and Safety (RAMS) measures are calculated according to guidelines suggested in ISA-TR84.00.02 [19] and IEC 61165 [18] standards, along with previous studies [1]. States are divided into two major categories: ‘up’ (or operational) and ‘down’ (or non-operational). In up states, the system is able to correctly run the safety functions. Up state is either all-OK initial state or any state with some tolerable failures. Down states are those in which system is not able to correctly run safety functions, either intentionally as in the fail-safe state or unintentionally as in unsafe (hazardous) state. System moves into fail-safe/unsafe state if there are intolerable number of dangerous detected/undetected failures present.

As soon as a dangerous failure is detected, system may either tolerate it (like the first detected failure in 2oo3 system) or enter into fail-safe state. On the other hand, if the failure is left undetected, system may inadvertently tolerate it (like first undetected failure in 1oo2 or 2oo3 systems) or enter into unsafe (hazardous) state.

PFH is defined as the average rate of entering into an unsafe state. For safety calculation purposes, repair transitions from unsafe states toward up states should not be considered [18]. Additionally, as in the case of reliability calculation, we also remove repairs from down fail-safe states to account only the effective safety when system is operational. Note that the repairs inside up states are not removed. All of the following Markov models are in the full form before repair removal. From the total failure rate of each system, $\lambda_{sys}$, only the hazardous part, $\lambda_H$, should be considered (as shown in Figure 5). By definition, PFH is calculated as the average of $\lambda_H$. The details of the following formula, required for decomposing $\lambda_{sys}$ into $\lambda_H$ and $\lambda_S$, is explained in the literature [14][$P_{HS}$: Probability of being in hazardous state, $P_{S}$: Probability of being in fail-safe state, $P'_{HS}$: derivative of $P_{HS}$ with respect to time]

$$\lambda_H = \frac{P'_{HS}}{1 - P_{HS}} \cdot \frac{P_H}{P_{HS}}.$$ 

Generally, probabilities of the system over time is described with the following set of differential equations:

$$P'_{1 \times n} = P_{1 \times n} \cdot A_{1 \times n},$$

where $P$ is vector of state probabilities over time, $P'$ is derivative of $P$ with respect to time, $n$ is number of states, and $A$ is transition rate matrix. Abbreviations used in Markov chains are listed in Table 3.

![FIGURE 5. Unsafe (hazardous) and safe failure rate.](image)

TABLE 3. Other abbreviations and symbols.

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>D</td>
<td>Dangerous failure, a failure which has potential to put system at risk (only such failures are considered in this paper).</td>
</tr>
<tr>
<td>C</td>
<td>Diagnostic coverage factor is the fraction of failures detected by online testing.</td>
</tr>
<tr>
<td>DD</td>
<td>Dangerous detected failure, a dangerous failure detected by online testing.</td>
</tr>
<tr>
<td>DU</td>
<td>Dangerous undetected failure, a dangerous failure not found by online testing.</td>
</tr>
<tr>
<td>CCF</td>
<td>Common cause failure (dependent failure).</td>
</tr>
<tr>
<td>$\lambda$</td>
<td>Total failure rate of component or system.</td>
</tr>
<tr>
<td>$\lambda'$</td>
<td>Independent failure rate of component or system.</td>
</tr>
<tr>
<td>$\lambda''$</td>
<td>CCF failure rate of component or system.</td>
</tr>
<tr>
<td>$\lambda_{DD}$</td>
<td>DD failure rate of component or system.</td>
</tr>
<tr>
<td>$\lambda_{DU}$</td>
<td>DU failure rate of component or system.</td>
</tr>
<tr>
<td>$s, d, t$</td>
<td>Number of redundancies (single, dual or triple).</td>
</tr>
</tbody>
</table>
independent (non-CCF) and can only be detected by self-testing. If failure is detected, next state is fail-safe, otherwise it is unsafe (Figure 6). Transition terms for 1oo1 system are:

$$k_i = 0, C_i = C_{\text{selftest}}$$

Transition rate matrix for illustrated Markov is as follows:

$$A = \begin{bmatrix} -\lambda_{DS} & \lambda_{DUS} & \lambda_{DDS} \\ \mu_{PT} & -\mu_{PT} & 0 \\ \mu_{OT} & 0 & -\mu_{OT} \end{bmatrix}.$$  

1oo2 Configuration. According to IEC 61508-6, 1oo2 system consists of two parallel channels which can both run the safety functions. However, only one dangerous-failure free channel is sufficient to keep the system safe. In this configuration, a single DU is tolerable which means that hardware fault tolerance (HFT) is equal to one. Since it is assumed that DD failures contribute to voting, then no DD failure is tolerated and system immediately enters into fail-safe mode (Figure 7). Generally, 1oo2 system has high safety against DU failures and low reliability against safe failures. Note that in Figure 7, since the failure in state (2) is undetected or hidden, the system seemingly works with two operational channels. Therefore, the system has similar behavior against DD failures in both state (1) and state (2). However, in fact the faulty channel is not counted as operational. Because state (2) is one step closer to unsafe state than state (1). Transition terms used for 1oo2 system are

$$\begin{align*}
\lambda_{DU}d &= 2[(1 - C') \cdot (1 - \beta_{1oo2})] \lambda_{PE} \\
\lambda_{DD}d &= [(1 - C') \cdot \beta_{1oo2}] \lambda_{PE} \\
\lambda_{DD}d &= \lambda_{DE}d + \lambda_{DD}d \\
&= (2 \cdot C' \cdot (1 - \beta_{1oo2}) + C' \cdot \beta_{1oo2}) \lambda_{PE} \\
\lambda_{DD}d &= \lambda_{DE}d + \lambda_{DU}d \\
&= \lambda_{DE}d + \lambda_{DE}d + \lambda_{DE}d + \lambda_{DE}d.
\end{align*}$$

Transition rate matrix for illustrated Markov is as follows:

$$A = \begin{bmatrix} -\lambda_{DD}d & \lambda_{DU}d & \lambda_{DD}d & 0 & \lambda_{DU}d \\ \mu_{PT} & -\lambda_{DD}d - \mu_{PT} & 0 & \lambda_{DD}d & 0 \\ \mu_{PT} & 0 & \mu_{OT} & 0 & 0 \\ \mu_{OT} & 0 & 0 & \mu_{PT} & \lambda_{DD}d \\ \mu_{PT} & 0 & 0 & 0 & \mu_{PT} \end{bmatrix}.$$  

2oo3 Configuration. Similar to 1oo2, 2oo3 is also capable of tolerating one DU failure, meaning hardware fault tolerance is equal to one. Besides, it has higher reliability (continuity of operation) due to being able to tolerate a single DD failure, similar to 2oo2 system (note that 2oo2 is not discussed here). Therefore, in literature, 2oo3 is known to have benefits of both 1oo2 and 2oo2 at the same time (as shown in Figure 8). However, due to more number of vulnerable channels (since

FIGURE 6. Markov model for 1oo1 (single) system.

FIGURE 7. Markov model for 1oo2 system.

FIGURE 8. Markov model for 2oo3 system.
total failure rate of all channels increases as number of channels increases), 2oo3 is neither as safe as 1oo2, nor as reliable as 2oo2. Note that, in such a system, we assume online repairing does not remove undetected failures. In Figure 8, the $\mu_{\text{OT}}$ transitions represent partial recovery (explained in Section III-D). Transition terms for 2oo3 system are ($\beta_2 = 0.3$)

$$\lambda_{\text{DU}}^t = 3 \cdot ([1 - C^t] \cdot (1 - 1.7 \beta)] \lambda_{\text{PE}}$$
$$\lambda_{\text{DD}}^t = ([1 - C^t] \cdot 2.4 \beta] \lambda_{\text{PE}}$$
$$\lambda_{\text{DD}}^t = \lambda_{\text{DD}}^t + \lambda_{\text{DD}}^t$$
$$= [3C^t \cdot (1 - 1.7 \beta] + C^t \cdot 2.4 \beta] \lambda_{\text{PE}}$$
$$\lambda_{\text{DD}}^t = \lambda_{\text{DD}}^t + \lambda_{\text{DD}}^t = \lambda_{\text{DD}}^t + \lambda_{\text{DD}}^t + \lambda_{\text{DD}}^t + \lambda_{\text{DD}}^t$$

Transition rate matrix for illustrated Markov is as follows:

$$A = \begin{bmatrix}
\lambda_{\text{DD}}^t & 0 & 0 & 0 & 0 \\
-\lambda_{\text{DD}}^t & \lambda_{\text{DD}}^t & 0 & 0 & 0 \\
\mu_{\text{PT}} & \mu_{\text{PT}} - \lambda_{\text{DD}}^t & 0 & 0 & 2/3 \lambda_{\text{DU}}^t \\
\mu_{\text{OT}} & 0 & -\lambda_{\text{DD}}^t & \lambda_{\text{DD}}^t & 0 \\
\mu_{\text{PT}} & \mu_{\text{OT}} & 0 & -\mu_{\text{PT}} - \lambda_{\text{DD}}^t & \lambda_{\text{DD}}^t \\
\mu_{\text{PT}} & \mu_{\text{OT}} & 0 & 0 & -\mu_{\text{OT}} \\
\mu_{\text{PT}} & 0 & 0 & 0 & 0 \\
- \mu_{\text{PT}} & 0 & 0 & 0 & 0 \\
\end{bmatrix}$$

V. EXPERIMENTAL RESULTS AND DISCUSSION

In this section, we investigate the influence of aforementioned parameters over PFH measure through solving Markov models of four configurations: 1oo1, 1oo2, and 2oo3 without/with partial recovery (2oo3 and 2oo3-PR). First, we give the initial state of these configurations corresponding to default parameter values for other RAMS measures - reliability and availability.

A. RELIABILITY AND AVAILABILITY

Reliability function, which is defined as probability of continuously staying operational, is depicted in Figure 9. Despite high safety level, the 1oo2 suffers from high rate of false-trips (transitions into fail-safe state), even more than the simple 1oo1. This follows from the fact that total failure rate of 1oo2 is around 2$\lambda$, and any single DD failure brings whole system into fail-safe state. This is the cost paid for having high safety with a simple architecture. Moreover, note that if partial recovery is not provided, more complex 2oo3 system is not much better than the others. After first DD failure, 2oo3 degrades to 1oo2 where reliability drops sharply even less than 1oo1. With partial recovery, the faulty PE with DD failure is quickly recovered largely reducing the probability of having two consecutive DD failures. Superiority of systems for operational availability at time = $\infty$ (steady state availability), at very low $\beta$ value (which is not practically achievable), is as expected (see Figure 10). However, as CCF rate increases, their order is swapped. This is due to the fact that staying more in operational states means higher probability of being exposed to DU CCFs and having a direct jump into unsafe state which takes considerable time to be recovered from. Nevertheless, availability values are almost same, except 1oo1 which is by far the lowest (1oo1 is not shown).

B. SAFETY SENSITIVITY ANALYSIS

In this section, we show the effect of variation in each of aforementioned parameters around defined default value, over PFH value. Mathematically, the following experiments show the partial derivations, $\partial \text{PFH}/\partial p$, where p is one of the safety parameters. SIL1-SIL4 safety levels are plotted by horizontal lines to show relative safety position. By such illustration of safety, designer perceives the distance of current design state from desired safety level. Besides, we also show a few pairs of relevant parameters in 2D-space. At initial states of configurations specified by default parameters, 1oo1 marginally could not achieve SIL2, while the rest are in SIL3 region.
As explained before, generally 1oo2 is the safer configuration, while 2oo3 has higher reliability.

**Sensitivity to λ_PE:** Figure 11 shows how safety is affected by different λ_PE values. Plots are almost linear in logy-logx plane with slope equal to one, which describe linear functions in y-x plane passing through the origin. In other words, $PFH = K \cdot λ_{PE}$. This is also understandable from Markov models, where most of the transition rates are linear function of λ_PE. Linearity implies that just by knowing the line slope which is achievable by having a single ($λ_{PE}, PFH$) point, and without solving the complicated Markov models or other techniques every time, safety of system can be tuned. For example, an order of magnitude (10X) improvement in λ_PE, results in shifting up one safety level (e.g., from SIL3 to SIL4).

**Sensitivity to β and β_2:** λ and β_2 are the indicators of mutual and trilateral isolation among PEs. It is a well-known fact that CCF failures have strong adverse effect on safety-critical systems. Figure 12 depicts how systems’ safety is affected by β variation. 1oo1 is independent from β as expected. One can observe from this figure that, for default parameters, it is quite difficult to get SIL4 through β improvement. Because by questionnaire method for β estimation (described in IEC 61508-6 [17]), β can hardly be estimated to be below 1 percent. A noticeable observation here is that similar to λ_PE, plots are almost linear in logy-logx plane. This linearity makes adjustment of safety by tuning β parameter easily, without needing to solve complicated mathematical models every time.

The β_2 is defined as a number in [0, 1] range, expressing part of β which is shared among all three PEs [8], where it is typically in 0.2-0.5 range. It is clear that 1oo1 and 1oo2 are independent from β_2. For a fixed β value, increase in β_2 leads to decrease of β_2 and obvious improvement of PFH. Therefore, to have a more meaningful analysis, instead of β, we fix the β_2. Figure 13 shows that the variation in β_2 has almost no effect over PFH. The explanation is that in 2oo3 system, any mutual or trilateral CCF failure leads to the same situation, fail-safe or unsafe state. This assumption has to be reminded from Section IV-A that CCFs are symmetric. In fact, this parameter affects the systems such as 1oo3 (not discussed here) in which the mutual CCFs can not defeat the redundancy, but trilateral CCFs can. One good design practice in such systems is to avoid sharing common resources among all PEs, such as communication links or power supply lines.

**Sensitivity to C_selftest:** According to formulas in Section III-C, self-testing is assumed to be equally effective
for both CCFs and non-CCFs. As shown in Figure 14, variation in this parameter can significantly affect the safety. For achieving SIL4 in the 1oo2 system, the $C_{\text{selfest}}$ has to be increased by 2 percent, while in 2oo3, it is more difficult, where at least 6 percent improvement is required (default of $C_{\text{selfest}} = 0.9$).

Sensitivity to $k_i$: $k_i$ is a constant which specifies the efficiency of comparison among PEs for detecting independent failures. Comparison is expected to be more efficient against non-CCFs than CCFs ($k_i = 0$ for 1oo1). In Figure 15, there is an unexpected behavior as $k_i$ has almost no sensible (or very small) influence on safety. The main reason for such observation is the absolute dominance of CCFs in above systems. More precisely, any DU CCF takes the whole system into unsafe state. However, two consecutive DU independent failures have to occur to cause the same situation which is far less probable. This is translated to an order of magnitude less influence of non-CCFs over safety. As a result, these systems seem to be rather insensitive to $k_i$.

One possible incorrect conclusion from this observation is to give up comparison for independent failures. But the fallacy is that whether a failure is dependent or not is not distinguishable before detection. As we will see, $k_i$ still has considerable effect on safety and as a result, comparison cannot be ignored. Since $k_c$ is usually as low as 0.1-0.4, a relaxed comparison mechanism that leads to $k_i$ value as low as $k_c$ is completely acceptable. Because it is enough to just have a reasonable value for $k_c$.

Sensitivity to $k_c$: $k_c$ is a constant which specifies the efficiency of comparison among PEs for detecting CCFs. In both 1oo2 and 2oo3 configurations, CCFs mostly have a larger negative influence when compared to independent failures. Because a single independent failure is tolerable in both cases. By the definition of CCF, comparison is not expected to be very efficient against CCFs ($k_c = 0$ for 1oo1). Nevertheless, experiments (as shown in Figure 16) show that $k_c$ still has a considerable effect over safety.

In case when one parameter is not sufficient to achieve the required safety level, simultaneous improvements on multiple parameters can be tried. Figures 17 and 18 show SIL regions in 2-D space while target safety is possible with values on SIL4 border lines.

Sensitivity to $\mu_{\text{OT}}$: Online repair which is invoked after online failure detection, is either employed when a single PE is not operational due to a DD failure (provided that partial recovery is available) or when DD failures are tolerated until whole system is in fail-safe state (if partial recovery is not available).
provided). Effect of repair rate in the former (only applicable to 2oo3 with partial recovery) is negligible. Since such repair does not reduce the number of DU failures. In the latter, effect is zero as expected (see Figure 19). Note that repairs from down states toward up states are not considered in PFH calculation (refer to Figure 5). In practice, this parameter is useful for adjusting reliability and availability.

**Sensitivity to $m_{PT}$**: Proof test and repair occurs periodically in long periods of time (at TI or test interval) to remove DU failures. It is either employed when whole system is in unsafe state or while a DU failure is being tolerated (as in both safe configurations: 1oo2 and 2oo3). In the former, its effect on PFH is zero, similar to online testing, since repairs from down states are removed in PFH calculation. In the latter, although number of DU failures are reduced, due to dominance of CCF rate, such improvement is not observed in safety (see Figure 20). A single DU CCF failure can defeat safety in both 1oo2 and 2oo3, while two consecutive DU independent failures have to occur for leading to the same situation.

**Sensitivity to Partial Recovery**: As illustrated earlier in Figure 9, if partial recovery (repair) is not provided for 2oo3 system, gain in reliability which is the main advantage of 2oo3 over 1oo2 is not significant. Therefore, in this case, usage of the more complex 2oo3 configuration is not logical. Unlike reliability, effect of partial recovery over safety (PFH) is negligible (shown in Figures 11, 12, 13, 14, 15, 16, 17, 18, 19, and 20). Note that, partial recovery removes DD failures affecting reliability due to less number of transitions into fail-safe state. On the other hand, PFH is mainly a function of DU failure rate.

**Sensitivity to $t_{D}$**: In online testing, time to detect a detectable failure ($t_{D}$), is the time between occurrence and detection of a failure. Equivalently, $\delta = 1/t_{D}$ is the frequency of online testing per hour. There is no direct reference to this parameter in IEC 61508 standard (except briefly for $\beta_{D}$ estimation), probably because in comparison to component failure rates, it is assumed to be negligible. In order to capture this parameter, an intermediate state is added to Markov chain for every DD failure transition (shown in Figure 21), in which the DD failure is temporarily considered as a DU failure. This additional state makes the Markov chain more complex. Therefore, due to Markov chain solution complexity, we only execute this for 1oo1 and 1oo2 configurations. Intermediate states can be one of three types, operational, fail-safe or unsafe, as other normal states. But, for PFH calculation, they are not absorbing, meaning the transition of online testing is not removed. They also contribute to decrease the safety by causing more possibility of transition into unsafe states. Based on experimental results shown in Figure 22, we can observe that if $\delta < 0.1$ per hour, safety level is slightly affected, while if $\delta < 0.01$ per hour, the effect is significant.

**VI. APPROXIMATE RATE CALCULATION ON MARKOV CHAIN**

Our experimental results show the significance of $\beta$-factor in safety systems. Considering this fact and the $\Omega(n^2)$ runtime complexity, we use an approximate rate calculation on Markov chain. In this method, we approximate the rates of transitions between states in the Markov chain. This approximation reduces the computational complexity of the analysis.

**FIGURE 19.** Effect of online repair rate variation over safety (refer to Table 2 for fixed parameter values).

**FIGURE 20.** Effect of proof-test rate variation over safety (refer to Table 2 for fixed parameter values).

**FIGURE 21.** An intermediate state is added to Markov chain, in which a DD failure has not been detected yet.

**FIGURE 22.** Effect of $\delta (=1/\delta)$ variation over safety (refer to Table 2 for fixed parameter values).
required for Markov transition (matrix multiplication formula given in Section IV-C, where \( n \) is the number of states), we propose a method for simplifying complex Markov chains into simpler ones. In this way, a quick and approximate failure rate can be calculated. Although CCF transitions have smaller rate value (multiplied by \( b \)), but due to jumping over several states, their impact on the system failure rate is decisive.

In the simple models depicted in Figure 23 (without online testing), two and three consecutive component failures move the system into fail state, while a single CCF has the same consequence (they are called as CCF2 and CCF3, depending on the number of jumps). In such a setting, it is desirable to compare the share of CCF and non-CCF transitions in the total failure rate. By using the conventional reliability formulations, failure rate is calculated for three scenarios: 1) full model, 2) keeping only the CCF transition, 3) keeping only non-CCF transitions. In the results shown in Table 4, as \( \beta \) increases, share of CCF transitions also increases. However, even at a smaller \( \beta \) value, order of the CCF rate is comparable to accurate result. While even solving such simple Markov chains requires computer applications, scenario 2 gives an immediate result in both models, where system failure rate is equal to CCF transition rate.

Following this observation, we propose a simplification technique for complex models. In this way, only the paths from start to fail state which include a CCF transition are preserved, while transitions or states not included in these paths are removed. The idea is explained on a simple Markov chain for a 1o03 system shown in Figure 24. Using the same model as shown in Figure 2, transition terms are

\[
\begin{align*}
\lambda_s &= \lambda_{PE} \\
\lambda d &= 2(1 - \beta) \cdot \lambda_{PE} \\
\lambda t &= 3(1 - 1.7\beta) \cdot \lambda_{PE}
\end{align*}
\]

Starting from state (1), there are four different paths that lead to the fail state, namely, 1-4, 1-2-4, 1-3-4, and 1-2-3-4, where shorter paths have longer CCF jumps and more share in PFH value. To model this, we start with an empty Markov without any transition, and we add these paths one by one (repair transitions are preserved). Solutions to these three approximate Markov chains and the main complete one are illustrated in Figure 25. As can be seen, even the simplest chain which includes only a single edge, provides a suitable approximation to estimate PFH and SIL level. In the simplest case, there is no need to solve the Markov as it is obvious that \( PFH = \lambda^3 t \).

**VII. CONCLUSION**

In this work, we analyzed the sensitivity of system safety to some critical design parameters in two basic multi-channel safe configurations, 1o02 and 2o03, where a 1o01 system is used as baseline. All configurations have been modeled by Markov chains to examine at which safety integrity level (SIL) they stand, and how distant they are from the target.
Hence, each safety parameter’s contribution on safety can be understood. Through these measurements, instead of blindly improving an unsafe system, designers can make an informed decision to select the most appropriate parameter for improvement. A significant presence of Common Cause Failures, is an important assumption for analyses in this study. Through experiments, we showed that there is a linear relationship between safety (PFH) and two parameters: $\lambda_{PF}$ and $\beta$, in logy-logx plane. We also observed that parameters which have considerable effect on CCF rate are more appropriate candidates for safety level enhancement. These include $\lambda_{PE}$, $\beta$, $C_{selftest}$, and $k^e$. Additionally, we propose a method for simplifying Markov chains in PFH calculation which can largely reduce the complexity to get an approximate result.

ACKNOWLEDGMENTS

This research is supported in part by TUBITAK grant 115E835, TUBITAK Teydeb 1501 program grant 3140492 and a grant from the Turkish Academy of Sciences. This study is an extension of our previous work [2] providing additional safety design parameters, illustration of reliability, availability, and initial unsafe state of systems with respect to SIL1-SIL4 levels in 2D plane, and a new approach for simplifying Markov chain.

REFERENCES


HAMZEH AHANGARI received the BS degree in computer hardware from the Sharif University of Technology, Iran and the MS degree in computer architecture from the University of Tehran, Iran. He is currently working toward the PhD degree in computer engineering at Bilkent University, Turkey. His research interests include reliability, safety, reconconfigurable architectures, and high performance computing.

FUNDA ATIK received the BS degree in computer engineering from Bilkent University. She is currently working toward the MS degree at Bilkent University and her supervisor is Dr. Ozcan Ozturk. Her research interests include parallel computing, GPUs and accelerators, and computer architecture.

YUSUF IBRAHIM OZKOK received the BS degree from Istanbul Technical University and the MS degree in electrical and electronics engineering from Middle East Technical University. He is employed as lead design engineer with Aselsan Defense System Technologies Division. He has been involving design of mission critical and safety critical embedded systems for about 15 years.

ASIL YILDIRIM received the BS and MS degrees in electrical and electronics engineering from Middle East Technical University. He is a senior software engineer with Aselsan Defense System Technologies Division. He is involved in development of safety critical embedded systems as embedded software engineer.

SERDAR OGUZ ATA received the BS degree in electrical and electronics engineering from Middle East Technical University and the MS degree in computer science from the University of Freiburg. He is currently working toward the PhD degree at Middle East Technical University. He is a software engineer with Aselsan Defense System Technologies Division. He is involved in development of safety critical embedded systems.

OZCAN OZTURK has been on the faculty at Bilkent since 2008 where he currently is an associate professor in the Department of Computer Engineering. His research interests include the areas of cloud computing, GPU computing, manycore accelerators, on-chip multiprocessing, computer architecture, heterogeneous architectures, and compiler optimizations. Prior to joining Bilkent, he worked at Intel, Marvell, and NEC. He is a member of the IEEE.