The effect of diagnostic and periodic proof testing on the availability of programmable safety systems

WOLFGANG VELTEN-PHILIPP
Automation, Software, Information
TÜV Rheinland
Bienwaldstr. 41, 76187 Karlsruhe
GERMANY
http://www.tuv.com

MICHEL HOUTERMANS, Ph.D.
Safety Management & Consultancy
Risknowlogy
Brunner bron 2, 6441 GX, Brunssum
THE NETHERLANDS
http://www.risknowlogy.com

Abstract: - The purpose of this paper is to show the effect diagnostic and periodic proof testing have on the availability of the safety function carried out by programmable electronic systems. For three different architectures the influence of the diagnostic coverage, the proof test coverage, and the proof test interval on the probability of failure on demand are determined. Performance indicators are used to express this influence and show the effect.

Key-Words: - Safety, reliability, safety systems, diagnostics, proof tests, 61508, 61511

1 Introduction
Safety systems carry out one or more safety functions. Each safety function consists of a sensing element, a logic solving element and an actuation element. Typical sensing elements are for example sensors, switches or emergency push buttons. The logic solving element is usually a general-purpose safety computer which can carry out several safety functions at once. Valves, pumps or alarms are typical actuating elements of a safety function. The performance of these safety functions is determined by several design parameters of the individual components of the safety functions. In [1] the following design parameters of the safety function are identified as the

- Architecture
- Hardware failures
- Software failures
- Systematic failures
- Common cause failures
- Online diagnostics, and
- Periodic test intervals.

The performance of a safety function can be expressed as the probability of failure on demand (PFD) and the probability of fail safe or spurious trip. The PFD value is one requirement to meet the safety integrity level of the IEC 61508 standard [2]. For the PFS value there are currently no requirements in the international safety world, although end-users of safety system require an as low as possible PFS value.

The purpose of this paper is to show the effect that the above mentioned design parameters, namely online diagnostic and periodic proof testing, have on the performance of the safety function in terms of the PFD. For three different architectures the influence of the diagnostic coverage, the proof test coverage, and the proof test interval on the PFD are determined. A performance indicator is used to express this influence and show the effect.

2 Diagnostic test versus periodic proof test
Before we can discuss the influence of diagnostics tests and periodic proof tests on the performance of the safety functions we first need to explain the difference between the two. Actually to diagnosis something means, "to distinguish through knowledge" [3]. In the safety world we use diagnostic tests to identify failures within the system which otherwise would not be revealed. In other words a diagnostic test is performed to find failures inside the safety system. But only revealing the
failures is not sufficient. Once a failure is found a decision needs to be made on what to do with that failure. Typical decisions made are shutdown or switch to degraded mode if the safety system has sufficient redundancy. But it is also possible to just notify an operator if the detected failure is not significant. A diagnostic test is not something that is clearly defined in IEC 61508. But everywhere it is mentioned it is clear that we deal with a test that is automatic.

The proof test is defined in IEC 61508 as well as in IEC 61511 [2,4]. IEC 61508 defines the proof test as follows:

“Periodic test performed to detect failures in a safety-related system so that, if necessary, the system can be restored to an “as new” condition or as close as practical to this condition”

IEC 61511 has a similar definition:

“test performed to reveal undetected faults in a safety instrumented system so that, if necessary, the system can be restored to its designed functionality”.

Thus in other words both diagnostic tests and periodic proof tests try to detect failures inside the safety system. At first sight there seems to be no difference yet the actual difference is quite an important one. Note 3 of paragraph 7.4.3.2.2 of IEC 61508-2 explains that a test is a diagnostic test if the test interval is at least a magnitude less than the expected demand rate. Based on this extra information we can conclude that in theory a test to detect failures is called a diagnostic test if the test is carried out automatically and more often than a magnitude less than the expected demand rate. In all other cases we can refer to a test as a proof test.

In practice we define a test as a diagnostic test if it fulfills the following three criteria:

1. It is carried out automatically (without human interaction) and frequently (related to the process safety time considering the hardware fault tolerance) by the system software and/or hardware;
2. The test is used to find failures that can prevent the safety function from being available; and
3. The system automatically acts upon the results of the test.

A typical example of a diagnostic test is a CPU – Test, a memory check or program flow control monitoring. A proof test on the other hand is a test which is carried out manually and where it is often necessary to have external hardware and/or software to determine the result of the test. The frequency of the proof test is much longer than the process safety time and magnitudes bigger than the period chosen for diagnostic tests. A typical example of a proof test is full or partial valve stroke testing. Partial valve stroke testing is seldom carried out without some form of human interaction (in other words we need to depend on the human to carry out the test, to determine the actual results of the test and/or to take the appropriate action based on the results) and often needs additional equipment to be carried out.

The advantage of a diagnostic test over a proof test is that failures can be detected very quickly. If a proof test is carried out once in three months then there is a possibility that the safety function is running with a failure for three months before we find out about it. With diagnostics we often know about the problem within milliseconds and thus can repair the failure very quickly. On the other hand though good diagnostics require more a complicated design and additional hardware and software build into the system. This additional hardware and software is often difficult to build, and costs extra.

There is another important reason to make a distinction between diagnostic tests and periodic proof tests. IEC 61508 requires the calculation of the safe failure fraction for subsystems. The safe failure fraction of a subsystem is defined as the ratio of the average rate of safe failures plus dangerous detected failures of the subsystem to the total average failure rate of the subsystem, see formula below.
A high safe failure fraction can be accomplished if we either have a lot of safe failures (detected or undetected does not really matter for the SFF) or if we can detect a lot of the possible dangerous failures. Only failures detected by diagnostic tests can be accounted for in the safe failure fraction calculation. Failures detected by periodic proof tests cannot be accounted for in the safe failure fraction calculations. This is logical of course, as we do not want to count on humans to carry out our safety.

3 Architectures
To show the effect that diagnostic and proof tests have on safety functions we introduce three common safety system architectures. The presented architectures are oversimplified but represent common structures that are used to implement safety functions. In practice these systems are much more complex and can consist of many more components. The four basic architectures presented can be characterized by their redundancy and voting properties, i.e., as XooY (i.e., “X out of Y”) and are
- The 1oo1 architecture;
- The 1oo2 architecture;
- The 2oo3 architecture; and

Each architecture consists of one or more sensors, one or more logic solvers, and one or more actuating elements. The following paragraphs will explain the three different architectures in more details.

3.1 The 1oo1 Architecture
The 1oo1 architecture is the simplest safety system around and consists of one single channel (see Figure 1). If any of the components within a channel fails dangerously the safety function cannot be executed anymore. If any of the components fail safe the safety function will be executed and a spurious trip will result.

3.2 The 1oo2 Architecture
The 1oo2 architecture, see Figure 2, consists of two channels, where each channel can execute the safety function by itself. If one channel fails dangerously the other channel is still able to execute the safety function and thus the safety system is still available.

If one channel has a safe failure the safety function will be executed and a spurious trip will follow.

3.3 The 2oo3 Architecture
The third architecture often used in practice is a 2oo3 voter structure, see Figure 3. This architecture consists of three independent channels. Each channel can carry out the safety function. The execution of the safety function requires two equal votes from the available channels. That means the safety function is executed if two channels have the same result. In other words we need two dangerous failures in two channels before the safety systems fails. We also need to safe failures in two channels before the safety function is executed and a spurious trip results.

4 Evaluation Procedure
The procedure used to calculate the PFD is outlined in detail in [1] and is in short as follows. A functional block diagram of the hardware of the safety system is drawn. For each element of the block diagram the typical failure modes are listed. The effect on system level is determined for each of them. This information is used to construct the reliability model, which is in this case a Markov model [5,6]. The last step in the procedure includes the quantification of the models by adding the failure and repair rates of the different

5 Reliability Data
One of the objectives of this study was to make sure that it was possible to compare the results for the different architectures. The actual value of the outcome is not so important as we are more
interested in the relative results. All calculation studies are carried out with these values unless otherwise noted in the specific study (Reference model).

Table 1. Default reliability data

<table>
<thead>
<tr>
<th>Component</th>
<th>Sensor</th>
<th>Logic</th>
<th>Actuator</th>
</tr>
</thead>
<tbody>
<tr>
<td>Failure rate [/h]</td>
<td>35E-6</td>
<td>15E-6</td>
<td>50E-6</td>
</tr>
<tr>
<td>Safe failures [%]</td>
<td>50</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>Diagnostic coverage [%]</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Online repair [h]</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>Repair after spurious [h]</td>
<td>24</td>
<td>24</td>
<td>24</td>
</tr>
<tr>
<td>Periodic proof test interval [y]</td>
<td>none</td>
<td>none</td>
<td>none</td>
</tr>
<tr>
<td>Proof test coverage [%]</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Mission time [h]</td>
<td>87600</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Common cause 2 failures [%]</td>
<td>0.0225</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Common cause 3 failures [%]</td>
<td>0.00225</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

6. Performance Indicators

In order to examine the results and the effects of changing the diagnostic, proof test coverage and parameters performance indicators are introduced. A performance indicator helps us understand what impact changing a parameter has on the PFD value of the system at hand. The performance indicator (PI) is calculated relatively simple. We change the parameter, for example the diagnostic coverage, stepwise from its minimum value to its maximum value. This will result in a PFD values changing according to the value of the parameter. To determine the impact of the parameter on the PFD value we calculate the change of the PFD relatively to the 50% value of the parameter. In case of diagnostic coverage 50% means 50% of the failures is detected by diagnostics. The 50% value is in this case the reference value that we use to normalize the PFD value to 1. To get an impression of the influence below 50% DC and above 50% DC we choose 25% DC and 75% DC for the PFD values to compare with.

The reason we do this is because we want to determine whether changing a parameter has a lot or only little effect on the PFD. In other words should we have in real life a system that does not meet our PFD value then we know which design parameter we have to address in order to make the changes that have the most influence and thus the fastest results. We don’t want to spend time and money on improving parameters that show little or no effect at all.

7. Calculation results

7.1 Calculation results with and without diagnostics – no proof test

In this paragraph we study the influence diagnostics has on the PFDavg values for the different system architectures as presented in paragraph 3.1. The diagnostics coverage is varied in the following percentages 0%, 25%, 50%, 75%, and 99%. The results are presented in the figure below.

Fig. 4. PFDavg in relation to diagnostic coverage

Next we calculate the performance indicator of the diagnostic coverage, i.e., how it influences the PFDavg of the different architectures. Figure 5 shows the change of the PFDavg for 25% DC compared to 50% DC and the change for 75% DC compared with 50% DC. The numbers are normalized with the 50% value.

Fig. 5. Performance indicators for variable diagnostic coverage

From Figure 4 and Figure 5 we can draw the following conclusions:

- The diagnostic coverage factor has a significant influence on the PFDavg value.
For all architectures counts that improving the diagnostic coverage from 0 to 100% will improve the PFDavg a factor of approximately 10 to 1000:
- The 1oo1 system is least sensitive to modifications of diagnostic coverage;
- The 1oo2 system is the most sensitive to modification of the diagnostic coverage factor. For low diagnostic coverage factors as well as high values;
- Both the 1oo2 and 2oo3 are one fault tolerant systems. The difference in performance can be explained be course of probability theory. Both systems need two failures in two different channels to loose the safety function. The 2oo3 system though has 3 possible combinations of two failures. But the probability of repair is equally fast in both systems;
- For all three architectures counts that making improvements has most effect above 50% diagnostic coverage.

In other words all architectures are sensitive for the diagnostic coverage factors and increasing the diagnostic coverage factor can make major improvements. The redundant structures are much more sensitive then the single structure. Diagnostic coverage really makes an impact on the PFDavg value when over 50% preferable over 75%.

### 7.2 Calculation results for variable proof test coverage

In this paragraph we study the influence the proof test coverage has on the PFDavg value for the different architectures as presented in paragraph 3.1. The proof test coverage varied in the following percentages 0%, 25%, 50%, 75%, and 100%. The results are presented in Figure 6.

Next we calculate the performance indicator of the proof test coverage, i.e., how it influences the PFDavg of the different architectures. Figure 7 shows the change of the PFDavg for 25% test coverage compared to 50% and the change for 75% compared with 50%. The numbers are normalized with the 50% value.

![Figure 6: PFDavg in relation to proof test coverage](image)

**Fig. 6.** PFDavg in relation to proof test coverage

![Figure 7: Performance indicators for variable proof test coverage](image)

**Fig. 7.** Performance indicators for variable proof test coverage

From Figure 6 and Figure 7 we can conclude the following:
- Also the proof test coverage helps improve the PFDavg but not as significant as the diagnostic coverage. An improvement with a factor of 10-100 is achievable;
- Just like with the diagnostic coverage the redundant architectures react better to proof test coverage improvements than the single architecture.
- The sensitivity concerning the PFDavg improvement is nearly the same for all architectures.

The PFDavg is less sensitive to proof test coverage compared to diagnostic coverage. This is understandable as diagnostic coverage gives almost very fast repair compared to proof testing. Failures detected with diagnostics are detected within seconds, while proof test can take 1 year or longer before they are carried out. Failures can thus exist much longer in the system. Therefore the PFD does on average not improve that much.
### 7.3 Calculation results with variable proof test interval

In this paragraph we study the influence of the proof test interval on the PFDavg value for the architectures as presented in paragraph 0. The proof test interval was varied from none to 3 months, 1 year, 2 years, and 5 years with a proof test coverage of 100%. The results are presented in Figure 8.

![Graph showing PFDavg in relation to the proof test interval](image1)

**Fig. 8. PFDavg in relation to the proof test interval**

Next we calculate the performance indicator of the proof test interval, i.e., how it influences the PFDavg of the different architectures. Figure 9 shows the change of the PFDavg for a proof test interval of 1-5 years and a proof test interval of 5-10 years. The numbers are normalized with the 5 years value.

![Graph comparing PFDavg for different proof test intervals](image2)

**Fig. 9. PFDavg for different proof test intervals**

Figure 8 and Figure 9 we can conclude that:
- There is a high sensitivity to proof test intervals varying approximately with a factor 10-1000 depending on the architecture;
- The redundant architectures benefit the most. The shorter the proof test intervals the more benefit. This is logical of course as the short the interval the more the proof test resembles a diagnostic test.
- The 1oo2 architecture benefits the most from frequent proof tests.
- There is almost no difference from a probability point of view to perform the proof test with a 5 or 10 years interval.

Just like the diagnostic coverage the proof test interval has a significant impact on the PFDavg value for all architectures. It is more important to carry out proof tests more frequently then with a high-test coverage. In other words even a simple but frequent proof test can help reduce the PFDavg value significantly. Thus is good news for partial stroke testing of valves. Especially with partial stroke testing of valves we do not know the actual proof test coverage but if done frequently it will still help significantly.

### 7. Calculation results

The purpose of this paper was to show the effect of diagnostic coverage, proof test coverage and the proof test interval on the PFDavg value for different safety architectures. To get more inside in the influence of these design parameters a performance indicator was introduced. The choice of diagnostic coverage and the proof test interval have the most influence on the PFDavg. The proof test coverage also improves the PFDavg but less significant then the other two parameters. The 1oo2 architecture gains the most benefits from while the 1o1 is the least sensitive. For redundant architectures counts that there is more chance on finding failures then there is for single architectures. Therefore they perform better in terms of improving the PFDavg. The authors are currently working on a more extended version of this paper taking among others into account more architectures and the PFS calculation per architecture.

### REFERENCES


