Proceedings of the Korean Nuclear Society Autumn Meeting
Taejon, Korea, October 2000

Software Reliability Estimation of Nuclear Critical Safety System Using Statistics of the Extremes

Dae Won Chung

Honam University
59-1, Seobong-dong, Kwangsan-ku, Kwangju-city, Korea 506-090

Abstract

Applying statistics of the extremes to the reliability estimation of nuclear safety softwares is proposed to improve alternately the reliability evaluation methodology. The software of nuclear safety functions such as reactor protection system and heat removal system frequently unable to produce pre-operation testing data and field experienced failure data for at least two reasons; the safety software is the one-of-a-kind for special purpose usage and it rarely fails in the testing. Therefore, the conventional Gaussian reliability model cannot be used for this purpose. Instead statistics of the extremes which is developed for analysis of rare events may be preferred to applying on this. This has such a advantage that it does not require a prior assumptions concerning the distributional form of failure history data but it concentrates on the tails of the data. It is presented in this paper the theoretical background and its applying methodology.

1. Introduction

During the recent years, the nuclear I&C systems performing the nuclear safety functions such as the reactor protection and shutdown and/or reactor heat removal, etc are trending toward replacing conventional analog signal processing systems with new digital technology of computer based systems. Although this kind of digital technology has significantly many technical advantages, it has not widely been applied to nuclear safety systems yet since there still are reluctances of the safety assessment and the licensing issues of these systems. In particular the software qualification and its reliability evaluation technique become new developing and challenging area of interest. The safety related software qualification should be achieved to minimize software design error by good developing practices of software validation such as non-nominal parameter analysis, failure mode analysis and hazard analysis, etc. Whereas, the software reliability is improved to minimize the safety system failure by validation testing of software such as fault injection test, stress testing and normal performance testing, etc. Fig. 1 shows the nuclear safety software developing flowchart in the most common practice. These novel activities on developing safety software are directly related to insuring the nuclear safety systems to perform the nuclear safety functions reliably since safety software faults are by far the hardest to predict effectively and the failures of a digital system are much affected by the software quality and unusual conditions of I/O signals.

The estimation of software reliability is required to assess the probability of fault-free operation of safety system within a specified period of time, which is based upon the failure history data of acceptance testing. Software reliability is generally defined as the probability of failure free software operation for a specified period of time in the specified environment[1]. In software reliability engineering, the reliability is usually estimated by the software failure intensity between failures to measure the frequency of the system failure as seen by the users. However, the reliability of the nuclear safety software can not be analyzed and estimated using conventional software reliability methods which are usually of Guassian probability model for the following reasons; Several software reliability models have been successfully applied to many commercial applications, but have the unfortunate drawback of requiring data from which one can formulate a model. It is the fact that nuclear safety software is frequently unable to produce such data since at the first, the software is frequently one-of-a-kind, and at the second, it rarely fails in the test. It means that nuclear safety system software is expected to pass every acceptance test and to produce precious little failure data during testing. Therefore, the other new methods for more reasonable and theoretical approach must be investigated to overcome this kind of situations.

Recently much work has been studied to improve the method of software reliability estimation for the critical safety systems[2-7]. However, the methods proposed by the most of literatures [2-7] have their own advantage and disadvantage to apply to critical safety systems and it is not desirable to discuss the their weak points in this paper. Furthermore, an estimation using statistics of extremes applying to reliability analysis of nuclear safety software may be one of the significant advantage.

This paper is to present a method to analyze software reliability estimation for the case when extensive testing reveals few or no failure records using statistics of the extremes[8-11]. The extreme statistics has been effectively applied for reliability analysis of rare events. This may be a good case to apply it on the software reliability analysis of nuclear safety systems

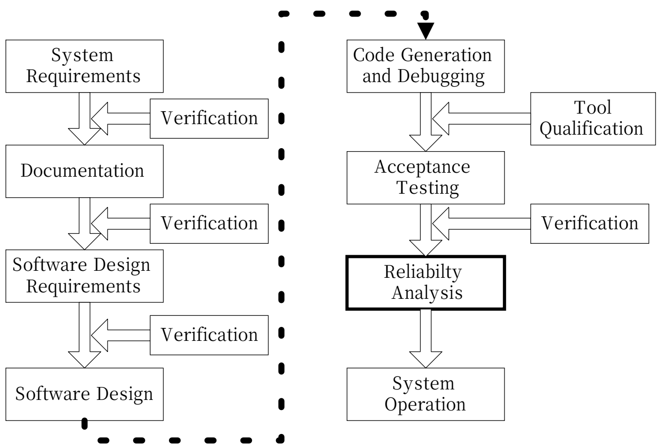

Fig. 1 Developing practice of nuclear safety software

since the failure data of nuclear safety software testing is little and rare event as discussed before. However, it is noted that this paper is not to discuss the safety functional requirements of nuclear safety softwares but is focussed on presenting the software reliability estimation methodology itself.

2. Overview of Statistics of the Extremes

Statistics of the extremes, which is based upon order statistics[8-11], is the study of the smallest or largest random variables that are obtained from a series of independent observations. It is shown in [10,11] that for most distributions as the number of observations approaches infinity, the distribution of either the maximum or the minimum values approaches one of three asymptotic forms[9-11].

A distribution F converges to an asymptotic form in its maximum extreme tail, if

$$H(x) = \lim_{n \to \infty} F_X^n(a_n + b_n x) \qquad \forall x \tag{1}$$

where a_n and b_n are non-unique series of constants. Similarly distribution L converges to an asymptotic form in its minimum extreme tail, if

$$L(x) = \lim_{n \to \infty} L_n(c_n + d_n x) = \lim_{n \to \infty} 1 - [1 - F_X^n(c_n + d_n x)]^n \qquad \forall x \tag{2}$$

where c_n and d_n are non-unique series of constants. These new functions, $H(x)$ and $L(x)$, are non-degenerative; i.e., in the limit, either of these functions constantly take the value one or zero, respectively. This property is called the stability postulate[10].

Using the stability postulate and Eq. (1), it has been shown in [2,4] that there are only three asymptotic distributions for the maximum extreme, which are;

$$H_{1,\gamma}(x) = \begin{cases} \exp(-x^{-\gamma}) & \text{if } x > 0,\ \gamma > 0, \\ 0 & \text{if } x \le 0, \end{cases} \tag{3}$$

$$H_{2,\gamma}(x) = \begin{cases} \exp[-(-x)^{\gamma}] & \text{if} x < 0,\ \gamma > 0 \\ 1 & x > 0 \end{cases} \tag{4}$$

$$H_{3,0}(x) = \exp(-\exp(-x)), \qquad -\infty < x < \infty \tag{5}$$

Similarly using the stability postulate and Eq. (2), there are only three asymptotic distributions for the minimum extreme tail, which are;

$$L_{1,\gamma}(x) = \begin{cases} 1 - \exp[(-x)^{-\gamma}] & \text{if } x \le 0,\ \gamma > 0, \\ 1 & \text{if } x > 0 \end{cases} \tag{6}$$

$$L_{2,\gamma}(x) = \begin{cases} 1 - \exp(-x^{\gamma}) & \text{if } x > 0 \\ 0 & \text{if } x \le 0,\ \gamma > 0 \end{cases} \tag{7}$$

$$L_{3,0}(x) = 1 - \exp[-\exp(x)], \qquad -\infty < x < \infty \tag{8}$$

The behaviour of a given distribution determines to which asymptotic family it belongs in its domain of attraction; i.e., if the distribution posses an exponentially decaying tail in the direction of the extreme of interest, it belongs to the Gumbel family; if the distribution possesses a polynomial decaying tail, then it belongs to the Frechet family; if the distribution is bounded in the direction of the extreme of interest which is a finite upper or lower value, then the distribution belongs to the Weibull family[8-11].

In applying statistics of the extremes to actual data, either a graphical or analytical technique may be used. Since the underlying distribution of the data is unknown within a priori, the method used to determine the domain of attraction of parent population is based on data samples. Graphically, the domain of attraction for a sample data set can be obtained using Gumbel type probability paper[9]. Once the data is plotted using appropriate weighting functions and the empirical cumulative distribution function (CDF) is generated, the curvature of the resulting plot determines the domain of the attraction. If the empirical CDF appears as a straight line, then distribution is a member of the Gumbel family. If the empirical CDF appears to be convex downward, then it is a member of the Frechet family. If the empirical CDF appears to be concave downward, then it is a member of the Weibull family. If it is determined that the asymptotic family is not Gumbel, then it is either Frechet or Weibull, and then the data is replotted on Frechet or Weibull paper, respectively. Since graphical techniques are subject to fitting, analytical techniques can be used to corroborate the results.

3. Methodology

3.1 Assumptions

Statistics of the extremes is used to model rare event data which is reflected in the tails of a given software failure probability distribution. Since extreme events are characterized by the tails and may be sensitive to small variations in the tails, a distribution chosen to model the entire range of failure events may not correctly model the tail events. It has been shown in [12] that fitting a separate distribution for the tail of the parent distribution may provide for a more accurate representation of an extreme event. The limiting form is independent of the central portion of the distribution[8]. The purpose of statistics model of the extremes is to provide a methodology for analyzing extreme values.

For the nuclear safety software in which testing yields little in any failure data, this failure data may be considered a rare event. A model can be developed that is independent of any a priori assumptions for the distributional form of this failure data. To utilize statistics of the extremes for the analysis of software when testing no failures, we may make following assumptions;

Assumption (1) : The occurrence of a software failure at the completion of the design and test phases is a rare event, and this failure data is found within the tail of the parent distribution.

Assumption (2) : The occurrence of each software failure is independent from the occurrence of any other given software failure.

Assumption (3) : The maximum number of software failures is limited to $D_{\max}$, where it depends on the software's frequency of execution and its expected lifetime. The entire sample space for the software actually contains a finite number of faults.

Assumption (4) : During the design and test phases, the software is thoroughly tested and modified to remove any detected faults. Use of good developing practice of software should be definitely required for this purpose.

Assumption (5) : It is too unrealistic to assume that every software fault is discovered and removed in the limit. i.e., If the softwares were tested for infinite time, it is assumed that there is at least one active fault in the system.

From these assumptions, it is implied that the statistics of the extremes model is an infinite failure model in which there is always at least one potential failure in the safety software. The maximum number of failures encountered in an actual system is $D_{\max}$. If $D_{\max}$ is large, then the occurrence of a software failure is not a rare event and the statistics of extremes model is no longer applicable. Instead the normal distribution(Gaussian) model of the Central Limit Theorem shall be applied for this case. The cutoff point that defines the theoretical boundary between these two is proposed around $D_{\max}=100$ in the reference [13]. However, this upper limit value may be adjusted, i.e., if software developer believe that the number of failure sample is too pessimistic, then $D_{\max}$ value may be adjusted to any specified value less than 100.

3.2 Analysis Method

Under these assumptions, it can be presumed that the number of software failures approaches the ranges from a finite lower bound of one to $D_{\max}$. Only the Weibull asymptotic form for minima should be considered because the number of the evaluated software failure approaches a finite lower value of 1 as the software is tested. In this method, analysis is independent of the number of times that software is actually demanded; i.e., this model predicts the maximum likelihood that Z software failures will occur in a nuclear safety system's lifetime.

3.2.1 Data Classification

In order to corroborate the decision that the data belongs to the Weibull family of minima, the hypothetical failure data is first analyzed using Gumbel probability paper[9] for minima. The data to be analyzed consists of the number of assumed software failures encountered ranging from 1 to $D_{\max}$ during the design and test phases. In order to demonstrate that the assumptions and subsequent software failure data belong to the Weibull family for minima, a value for $D_{\max}$ should be specified at first. The resulting plot which is referred to as the empirical CDF, reflects the CDF of the data; i.e., it demonstrates the probability that the number of software failures is less than or equivalent to a specific value.

The example result of this graphical analysis is shown in Fig. 2. In this example of our analysis, it was assumed to be $D_{\max}=100$. From this figure the curvature of the plotted data appears to be concave downward, which suggests the Weibull domain of attraction. Since this method is a little bit subjective fitting, the domain of attraction for the data can be further corroborated using analytical methods. Hence, the assumptions that led to the classification of the software failure data as belonging to the Weibull asymptotic family for minima are appropriate.

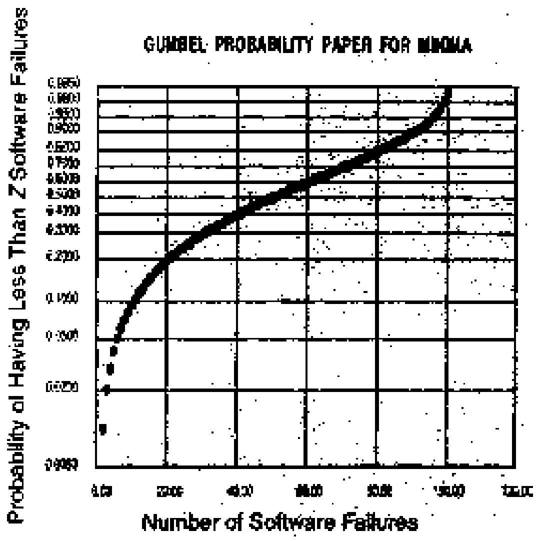

Fig. 2 Plot of software failure data on Gumbel probability paper [9]

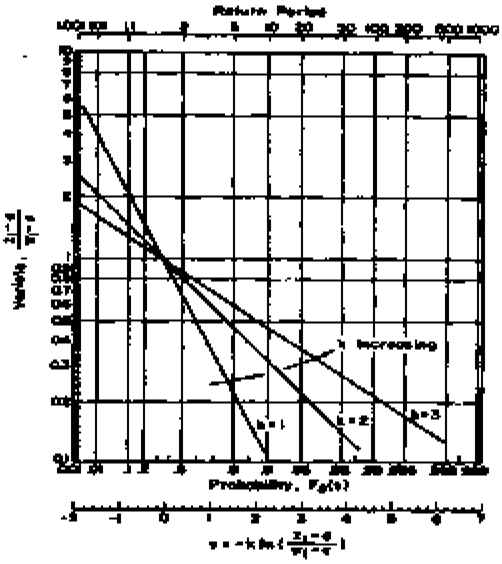

Fig. 3 Weibull empirical exceedance CDF for minima [10]

3.2.2 Data Analysis

Using the form for the Weibull asymptotic family[10], which is equivalent to the expression found in Eq. (7), a graphical based analysis using empirical CDFs can be performed to determine the software reliability estimation for an ultra-dependable system. The Weibull distribution is given in [10] as

$$F_S(s) = \exp(-\exp(-s)) \tag{9}$$

where

$$\begin{aligned} s &= -k\ln\left(\frac{x_1-\varepsilon}{w_1-\varepsilon}\right) = -k[\ln(x_1-\varepsilon) - \ln(w_1-\varepsilon)] \\ &= -\ln[-\ln[F_S(s)]] \end{aligned} \tag{10}$$

w_1 is the characteristic smallest value of the initial variate X, which is a measure of the central location of the smallest possible value, ε is the lower bound, and k is an inverse measure of dispersion of X_1. The graphical representation of Eq. (9) is shown in Fig. 3. The figure uses log-extremal probability paper in generating the empirical CDF of exceedance. The empirical CDF is a plot of the probability that the number of software failures encountered in a real application will exceed a specified value. Using the information provided from this exceedance CDF, the parameters for the Weibull distribution as defined by Eq. (9) and (10) can be extrapolated. the slop of the line representing the empirical exceedance CDF is described by Eq. (11);

$$k = -\frac{-s}{\ln(x_1-\varepsilon) - \ln(w_1-\varepsilon)} \tag{11}$$

The characteristic smallest value (w_1) occurs at $s=0$, hence this value can be extrapolated from the graph. Once this value is determined, then the slop of the line representing CDF as expressed in Eq. (11) can be calculated.

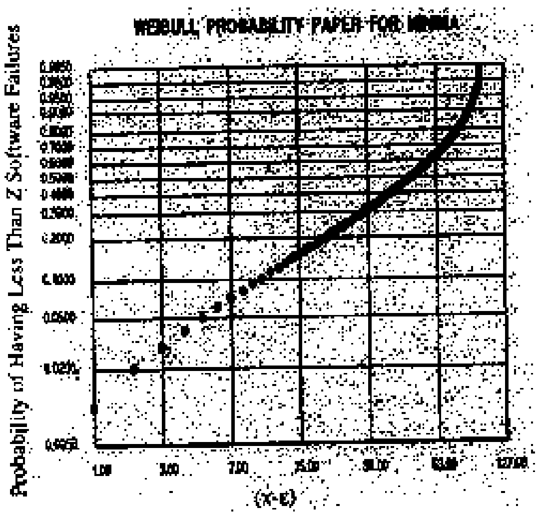

Fig. 4 Weibull empirical CDF for software failures [10]

4. Application Study

4.1 An Applicable Example

Assuming that a maximum of 100 software failure at one of the nuclear safety functions may occur (i.e, $D_{\max}=100$), the graphical representation of failure data is shown in Fig. 4. This graphic is empirical CDF of the data, and from this graph, it may be extrapolated that the characteristic smallest value occurs at $w_1=64$. i.e., $s\approx 0$, which occurs with a probability of 0.632, when $(x-\varepsilon)\approx 63$. Using Eq. (10) and Fig. 4, a value for s can be extrapolated. the probability of having less than $(x-\varepsilon)=60$ software failure is approximately 0.41 ($F_S(s)\approx 0.41$) when $x=61$. Hence, the value of s for $(x-\varepsilon)=60$ is;

$$s=-\ln[-\ln(0.41)]=-0.11473979 \tag{12}$$

Using Eq. (11) and the parameters used to derive the expression found in Eq. (12) , the inverse measure if dispersion is calculated;

$$k=-\frac{-s}{\ln(x_1-\varepsilon)-\ln(w_1-\varepsilon)}=-\frac{-0.11473979}{\ln(63)-\ln(60)} \approx 2.35 \tag{13}$$

and the resulting analytical expression for CDF from Eq. (9) is;

$$F_S(s)=F_{X_1}(X_1\leq x)=1-\exp(-\exp(-s)) \tag{14a}$$

Table 1 : Probability of Exceeding Z Software Failures Assuming $D_{\max} = 100$

Probability of Exceeding Z Software Failures	Z : Number of Potential Lifetime Software Failure					
	2	10	25	50	75	100
$P_{X_1}(X_1 \geq x)$	0.9999	0.9897	0.9017	0.5746	0.2323	0.0555

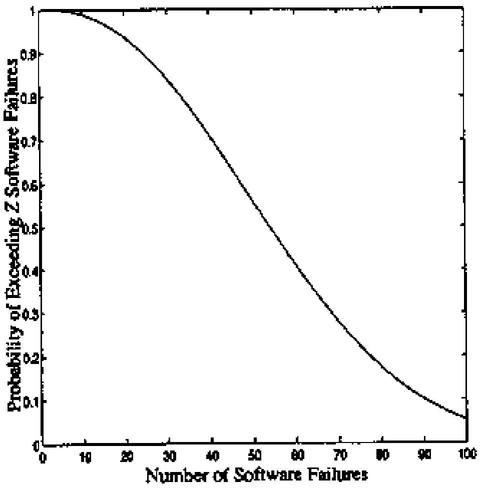

Fig. 5 CDF of Z or less Software Failures

$$= 1 - \exp\left[-\left(\frac{x-\varepsilon}{w_1 - \varepsilon} \right)^k \right] \quad (14b)$$

$$= 1 - \exp\left[-\left(\frac{x-1}{63} \right)^{2.35} \right] \quad (14c)$$

Eqs. (14) indicate the probability that the number of software failure is less than x. Another item of interest is the probability i.e, $P_{X_1}(X_1 \geq x)$ that the number of software failure exceeds a specified level. This exceedance CDF is plotted in Fig. 4 and some of the resulting probabilities are summarized in Table 1.

From both the graphic and the table, it is demonstrated that the likelihood of exceeding Z software failures diminishes as the number of software failures approaches 100 (i.e., $D_{\max} = 100$). Hence, the most conservative software reliability estimation for a given system has the greatest likelihood occurrence; i.e., if it is assumed that a system has 100 software failures, then probability of experiencing more than 100 failures is less than 0.0555 (it should be actually zero). The reason why this value is greater than zero is due to round-off errors during the plotting of the data.

Since the software failure estimation depends upon the number of times the software is used, the Weibull analysis must incorporate the software demand. Hence number of times that a given piece of software is the testing demand(N), is varied from 10,000 to 1,000,000 and software failure and success estimations are given by Eq. (15) and (16), respectively;

$$\textit{Software Failure Estimation} = \frac{Z}{N} = \hat{\theta} \quad (15)$$

$$\textit{Software Success Estimation} = 1 - \frac{Z}{N} = 1 - \hat{\theta} \quad (16)$$

The resulting software success (reliability) estimations are summarized in Table 2.

The probability [$P_{X_1}(X_1 \geq x)$] measured from Table 1 is actually less than the predicted value. i.e., it predicts the likelihood that more than Z software failures are contained within

Table 2 : Software Reliability Estimation Assuming $D_{\max} = 100$

N : Number of Test Demands	Z : Number of Potential Lifetime Software Failure				
	2	25	50	75	100
10,000	0.999800	0.997500	0.995000	0.992500	0.990000
100,000	0.999980	0.999750	0.999500	0.999250	0.999000
250,000	0.999992	0.999900	0.999800	0.999700	0.999600
500,000	0.999996	0.999950	0.999900	0.999850	0.999800
750,000	0.999997	0.999967	0.999933	0.999900	0.999867
1,000,000	0.999998	0.999975	0.999950	0.999925	0.999900

the system. As previously stated, the most conservative software reliability estimation, which assumed 100 software failure, is the least likely to be exceeded. Since the number of software failures seen in a system is modeled independently from the number of demands, the probability measure [$P_{X_1}(X_1 \geq x)$] depends only on the number of failure seen, which is assumed to be limit to 100 or less in this estimation; i.e., $P_{X_1}(X_1 \geq x)$ is independent of the number of software testing demands. Hence, its value is a constant reflecting the likelihood of M software failures occurring.

4.2 Comparison with Other Useful Models

In order to evaluate the accuracy of software reliability estimation, the comparative analysis is made for a hypothetical example. The models chosen for this comparison are illustrated for Parnas et al model[5] and the Miller et al model[6]. It was assumed that no prior test data exists for a given piece of software and that all available data exists from current testing, the various reliability models can be compared to demonstrate their differences. And it was also assumed that for our hypothetical example, there are one million test cases executed revealing three failures occurring in tests 3, 1175 and 919,992, then the various models can be exercised to demonstrate their differences.

4.2.1 Parnas et al Model

The probability of software failure in the Parnas et al model is simply based upon the number of test cases in first failure occurring in Eq. (17).

$$\hat{\theta} = \frac{1}{h} \tag{17}$$

where h is the number of test cases in which no failure has been encountered. In this reliability estimation, three separate calculation must be made because the estimation depends upon the number of test cases seen before there is a failure. i.e., the failure probability depends upon the sampling order of tests. Three estimation for the probability of failure of the given software are;

$$\hat{\theta}_3 = \frac{1}{3} = 0.3333 \tag{18a}$$

$$\hat{\theta}_{(1,175)} = \frac{1}{1,175-3} = 853.2e^{-6} \tag{18b}$$

$$\hat{\theta}_{(919,992)} = \frac{1}{919,992-1,175} = 1.088e^{-6} \tag{18c}$$

In this example, the software failure probability depends upon when the software failure is encountered, which is a function of the sampling order; i.e., the greater the number of tests encountered before failure, the lower the failure probability is predicted. Since this method is affected sensitively by the sampling order of the test cases, it can be either pessimistic or optimistic.

4.2.2 Miller et al Model.

This model uses a Bayesian approach with sampling replacement when testing reveals no failures in the current version of the software. The concepts and detail calculation methods are required to refer the reference [6]. The three estimation for failure probability are;

$$\hat{\theta}_3 = \frac{1}{3+2} = 200.0e^{-3} \tag{19a}$$

$$\hat{\theta}_{(1,175)} = \frac{1}{(1,175-3)+2} = 851.8e^{-6} \tag{19b}$$

$$\hat{\theta}_{(919,992)} = \frac{1}{(919,992-1,175)+2} = 1.088e^{-6} \tag{19c}$$

Since this method is also affected sensitively by the sampling order of the test cases, it can be either pessimistic or optimistic.

4.2.3 Statistics of the Extremes Model

In this model implementation, sampling order of the test cases is irrelevant to the estimation of software sucess/failure reliabilities. Rather, the estimation and its associated probability depend only on the cumulative failure data collected during real testing. Hence, the failure probability for the hypothetical software is;

$$\hat{\theta}_3 = \frac{3}{1,000,000} = 3.0e^{-6} \tag{20}$$

with an associated probability of occurrence of exceeding Z software failures, assuming that $D_{\max} = 100$, which derived in Eq. (14) is based upon the previous assumptions. From this equation, it is apparent that there almost certainly, with probability of 0.9997, will be more than 3 failures encountered during the software's lifetime, hence the failure probability($\hat{\theta}$) predicted from the experimental data is overly optimistic. However, if it can be hypothesized that assuming a total of 100 failures during the software's lifetime is overly pessimistic, then the value of $D_{\max} = 100$ can be set to a lower value.

In this example, if we hypothesized that the total number of software demands over the software's lifetime is ten times the number of test cases, then $D_{\max}$ can be set to a value of

30 (i.e., $D_{max}=30$). However, we also have to consider the safety factor for margin in this value, so the value is set to 60 (i.e., $D_{max}=60$). The resulting analytical expression for the failure probability that the number of software failure is less than x is determined in the same manner as presented in Section 4.1;

$$F_S(s)=F_{X_1}(X_1 \le x)=1-\exp(-\exp(-s)) \tag{21}$$
$$=1-\exp\left[-\left(\frac{x-\varepsilon}{w_1-\varepsilon}\right)^k\right]$$
$$=1-\exp\left[-\left(\frac{x-1}{38}\right)^{1.05}\right]$$

Table 3 : Probability of Exceeding Z Software Failures Assuming $D_{max}=60$

Probability of Exceeding Z Software Failure	Z : Number of Potential Lifetime Software Failure					
	3	10	15	30	45	60
$P_{X_1}(X_1 \ge x)$	0.956	0.802	0.704	0.471	0.311	0.205

The probability estimation is summarized in Table 3. Of more interest to the software developer is the probability of exceeding Z software failures for $D_{max}=60$ case. This information may also be compared with the result of Table 2.

From these analysis, it is clearly apparent that the probability of exceeding the number of failure encountered during testing is sensitive to the value selected for D_{max}. If this value selected as the best value, we can expect the more accurate extrapolation of probability distribution and can get more accurate reliability estimation. Therefore, this value must be specified based upon the testing experienced data for real application.

5. Conclusions

Assuming that a software failure is a rare event for the nuclear safety systems, the use of statistics of the extremes in software reliability analysis provides for an estimation of the software reliability that is dependent upon neither the choice of a distribution for the time-to-failure data nor the sampling order of the test cases. The use of statistics of the extremes for this analysis precludes the need of any a priori information on the distributions of the various parameters associated with the collected data since its probability of failure estimation is derived from the cumulative failure data. This provides for us to measure quantitatively the most likelihood that estimates the safety software reliability effectively and determines if the reliability estimation is too pessimistic or too optimistic in the whole life-time.

[References]

[1] ANSI/IEEE, "Standard Glossary of Software Engineering Terminology," STD-729-1991, ANSI/IEEE, 1991.

[2] J.C. Laplace, and M. Brun, "Critical Software for Nuclear Reactors; 11 Years of Field Experience Analysis," IEEE Nuclear proceedings, pp. 364-368, Sep. 1998.

[3] L. Pham and H. Pham, "Software Reliability Models with Time-Dependent Hazard Function Based on Bayesian Approach," IEEE Transaction on Systems, Man and Cybernetics, vol. 30, No. 1, Jan 2000. pp. 25-35.

[4] M. Bouissou, F. Martin and A Ourghanlian, "Assessment of a Safety-Critical system Including Software: A Bayesian Belief Network for Evidence Sources," IEEE Proceedings Annual Reliability and Maintainability symposium, 1999, pp. 142-150.

[5] Parnas, L. David, J. V. Schouwen, and S. P. Kwan, " Evaluation of Safety Critical Software," Communication of the ACM, vol. 33, no. 6, June, 1990, pp. 636-648.

[6] K.W. Miller, L.J. Morell, R.E. Noonan, SS.K. Park, D. M. Nicol, B. W. Murrill and J. M. Voas, "Estimating the Probability of the Failure When Testing Reveals No Failure," IEEE Transaction on Software Engineering, vol. 18, no.1, Jan. 1992, pp. 33-43.

[7] T.W. Keller and N.F. Schneidewind, " A Successful Application of software Reliability Engineering for the NASA Space Shuttle," Proceedings if the Eighth International Symposium on Software Reliability engineering, 1997, pp. 71-82.

[8] Castillo, Enrique and Jose Sarabia, "Engineering Analysis of Extreme Value Data; Selection of Models," Journal of Aterway, port, Coastal and Ocean Engineering, vol. 118, no. 2, Mac. 1992, pp. 129-146.

[9] E. J. Gumbel, **Statistics of the Extremes**, Columbia Press, New York, 1958.

[10] A. H. Ang, and W. H. Tang, **Probability Concepts in Engineering Planning and Design**, volume II, John Wiley and Sons, New York, 1984.

[11] Castillo, Enrique, **Extremes Value Theory in Engineering**, Academic Press, Inc., San Diego, California, 1988.

[12] J.H. Lambert, N. C. Matalas and D. Li, "Selection of Probability Distributions and Characterizing Risk of Extremes Events," Risk Analysis, vol.14, no.5, May 1994, pp.731-742

[13] A. Papoulis, **Random Variables and Stochastic Processes**, McGraw-Hill, Inc., New York, 1991.

[15] N. F. Schneidewind, "Reliability Modeling for Safety Critical Software," IEEE Transactions on Reliability, vol. 46, no. 1, March, 1997, pp. 88-98.

[16] N. F. Schneidewind, "Software Reliability Model with Optimal Selection of Failure Data," IEEE Transactions on Software Engineering, vol. 19, no. 11, Nov., 1993, pp. 1095-1104.