Dependability Modeling and Evaluation of Recovery Block Systems

Katerina Goševa – Popstojanova

Faculty of Electrical Engineering

91000 Skopje, Macedonia

Aksenti Grnarov

Faculty of Electrical Engineering

91000 Skopje, Macedonia

Abstract

The paper presents performance modeling and evaluation of recovery block systems. In order to produce dependability model for complete fault tolerant system we consider the interaction between the faults in the alternatives and the faults in the acceptance test. The study is based on finite state continuous time Markov model, and unlike previous works, we carry out the analysis in the time domain. The undetected and total failure probabilities (safety and reliability), as well as the average recovery block execution time expressions are obtained. Derived mathematical relations between failure probabilities (i.e. reliability and safety) and modeling parameters enable us to gain a great deal of quantitative results.

1 Introduction

Computing systems are used in increasingly complex situations, hence system complexity itself becomes one of the major barriers to achieve required high level of reliability. A great part of this complexity is in the software, so it is a critical part of any high reliable computing system. Despite of fault prevention techniques (precise specifications, design methodologies, structured programming techniques, proving, testing, etc) which may have been used, a complex software system will always contain design faults when it is put into operation. Therefore, the importance of software fault tolerance can only increase.

The tolerance of software design faults relies on the application of design diversity approach in which components are independently designed to meet the same system requirements [8]. The recovery block (RB) scheme [1],[10],[18] for achieving software fault tolerance by means of stand by sparing consists of a primary alternate, a list of supplementary alternates and an acceptance test. A typical software structure to implement a recovery block was proposed by Randell [18]:

ensure Acceptance test
by Primary alternate
else by Alternate 1
⋮
else by Alternate N
else error

The primary alternate is one which is intended to be used normally to perform the desired operation. Other alternates might perform the desired operation in some different manner, preferably more simple. The acceptance test is evaluated on exit from any alternate to determine the acceptability (rather than the complete correctness) of the results produced by an alternate. When the RB is entered, the primary alternate is executed. If the acceptance test is passed, any further alternates are ignored and the control is transferred to the statement following the recovery block. Otherwise, the state of the process is restored to that current just before the entry to the primary alternate (backward error recovery), and a further alternate (if one exists) is executed. This process is repeated until either the result from an alternate is accepted or there are no more alternates to execute. When the

0-8186-4010-3/93 $03.00 © 1993 IEEE

last alternate fails to pass the acceptance test the entire RB is regarded as failed and a high level (global) recovery is performed. The execution of acceptance test, alternative blocks, and state recovery may cause a considerable run – time overhead.

2 Background

The dependability of computer systems has become a critical attribute for success of computer based applications. The notion of dependability means to perform with little probability of unexpected behavior (i.e. we can depend on computer system service or trust it). Dependability enables various concerns to be subsumed within a single conceptual framework and thus includes such attributes as reliability, availability, safety and security. We are going to consider the reliability and safety attributes of the RB system. According to widely accepted definition, the reliability is the probability of accomplishment of a function under specified environmental conditions and over a specified time. In the absence of repairs from a system failure the availability and reliability are equivalent. Safety is the probability that no catastrophic accidents will occur during system operation under given conditions for a specified period of time. It is now widely accepted that safety and reliability are not synonymous. In general, reliability requirements are concerned with making a system failure – free, whereas safety requirements are concerned with making it accident – free. Conversely, if the system fails, but fails to a safe state, it is certainly not reliable but it is safe.

There are two main approaches to dependability analysis: testing and modeling [14]. Results of testing are more believable than those from modeling. It is well known, however, that testing can only establish the presence of errors but cannot assure their absence. Also, for highly dependable systems testing is not always feasible and tend to be extremely expensive to develop and run to obtain statistically significant results. For complex fault tolerant systems dependability modeling and prediction have become an integral part of the system design process. Thus, an early analysis during system development is possible and it provides information regarding whether the current design will be able to attain the dependability requirements, and which parts of the design are the weak points with respect to dependability. On the basis of these models tests may be designed in order to prove the assumptions of the model, giving a cost effective method for dependability validation of the system. Interaction between modeling and experimentation can help us both in understanding the problems and yielding specific numerical measures.

A number of papers devoted to the dependability analysis of software fault tolerance approaches have appeared in the literature. We are going to give a brief overview of the papers concerning RB systems.

Grnarov, Arlat and Avizienis [7] have developed a general model for unified interpretation of the software fault tolerant strategies. In order to determine the average segment processing time, the queueing theory is used. The reliability models are combinational, considering the probability of correlated errors. Thus, the overall system reliability is evaluated as a number, i.e. the time is not incorporated in the expression of the reliability. Laprie [9] has developed Markov model for RB system with two alternatives and acceptance test. The reliability is modeled asymptotically by a homogeneous Poisson process. An equivalent failure rate is approximately derived as a reciprocal value of the mean time to failure of the presented Markov chain. Scott, Gault and McAlister [16] have proposed the combinational software fault tolerant reliability models based on the probability axioms. The RB reliability model is obtained from the event tree. The proposed dependent reliability model can be used to predict reliability only if the joint probabilities could be found. Mulazzani has applied the model presented in [13] to different software fault tolerant techniques. The estimation of the probabilities of good result, faulty result and no result is done by a simple combinational model. The constant value is attached to each alternate result class, as well as to the correlation of the faults of different alternates. Cha [3] has

combined the software fault tolerant models presented in [13] and [16]. His combinational model allows the independence between successes of any alternate and the acceptance test. The probabilities of common failures are also included. Arlat, Kanoun and Laprie [2] have developed dependability models (encompassing reliability and safety issues) of the software fault tolerant approaches. The reliability of RB with two alternates is modeled asymptotically by homogeneous Poisson process. An equivalent failure rate is derived by using the departure rate and the probability of failure obtained from the embedded discrete time Markov chain.

3 Recovery block modeling

Software faults can be manifested only when the software is executed [15], so we shall consider the execution process and the fault manifestation. Our study is based on finite state continuous time Markov model because important interdependence and dynamic relationship among system components are easily lost when using simple combinational models. Unlike previous works, we carry out the analysis in the time domain. That is, we are able to determine the reliability and safety at any given value of the execution time, like in our previous papers concerning the N version programming reliability modeling [5],[6].

The methodology used for RB modeling is based on the identification of the possible types of faults and analysis of the behavior following the fault activation. Two classes of faults are distinguished:

- *independent faults* in any alternate or in the acceptance test
- *related faults* among several alternates or between an alternate and the acceptance test.

The related faults among alternates have no influence on the RB because the absolute decision is taken for each alternate [2]. Hence, we consider only the interactions among the independent faults in any alternate, independent faults in the acceptance test, and the related faults between the alternates and the acceptance test.

If there are no active faults neither in the alternate nor in the acceptance test the correct result from the alternate passes the acceptance test. The following three error types result from the activation of faults:

1. Incorrect result from any alternate is accepted by a faulty acceptance test (an erroneous result is delivered);
2. Any alternate produces an incorrect result and the acceptance test labels the result as incorrect (no acceptable result is identified by the acceptance test);
3. Any alternate produces correct result, but the acceptance test erroneously determines that the result is incorrect (no acceptable result is identified by the acceptance test).

The ability to detect a failure (errors of type 2 and 3) may be an important consideration, in the sense that an undetected failure (an error of type 1) may have and generally has, catastrophic consequences.

The state recovery error is not included because the design of recovery mechanism is sufficiently simple that standard hardware design practices can ensure that there are no residual faults in its design [18]. The correct operation of the recovery mechanism is necessary to enable the operation of RB system and and we assume that its operation will be reliable.

We have made the assumptions that:

- the number of the alternate failures in a given time period follows a Poisson distribution (i.e. the failure intervals follow an exponential distribution) with parameter λ;
- alternate's execution time is exponentially distributed random variable with parameter μ_A;
- the service rate of the global recovery is $\mu_R = \frac{\mu_A}{R}$.

These assumptions guarantee that a finite state continuous time Markov chain can be used for dependability modeling of the recovery block system.

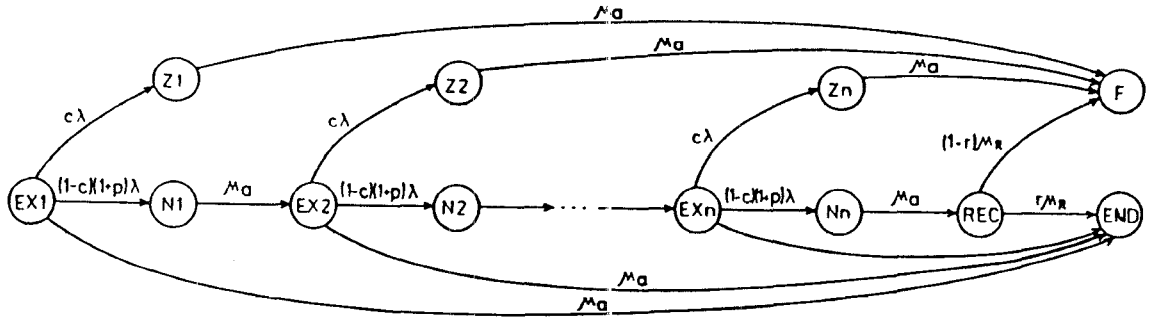

Figure 1: Markov model of the recovery block system

It is obvious that an understanding of fault tolerant diverse systems must depend on an understanding of the behavior of the components from which they are constructed. It is assumed that the alternate failure rates would be estimated before they are integrated into the RB system, for example using the software reliability models [11],[12],[15],[17].

The fault tolerant approach clearly depends on the degree to which diversity of the failure behavior can be achieved. We shall analyze the failure behavior of the RB system, on a particular input, at the end of the execution. The most significant issue of diversified software in operation concerns the types of failures (similar or distinct) that result from the activation of faults. Similar coincident failures mainly originate from related design faults, although in some rare cases it is also possible independent design faults to produce coincident failures and give either distinct or similar errors. It is worth noting that the detailed analysis of the relationship between classes of faults and failures would result in a further increase in the number of parameters of the model.

We define two failure rates:

- undetected failure rate $\lambda_{s_{AL-AT}}$ (coincident and similar failures in any alternate and the acceptance test);

- detected failure rate $\lambda_{d_{AL}} + \lambda_{d_{AT}}$ (separate or coincident distinct failures in any alternate and the acceptance test).

Let c $(0 \leq c \leq 1)$ be the fraction of the alternate failures that are determined as correct results. Then

$$\lambda_{s_{AL-AT}} = c\lambda$$

$$\lambda_{d_{AL}} = (1-c)\lambda$$

$$\lambda_{d_{AL}} + \lambda_{d_{AT}} = (1-c)\lambda\left[1 + \frac{\lambda_{d_{AT}}}{(1-c)\lambda}\right].$$

If we define $p = \lambda_{d_{AT}}/[(1-c)\lambda]$ then the detected failure rate is $(1-c)(1+p)\lambda$, where $p \geq \lambda_{d_{AT}}/\lambda$.

3.1 Dependability modeling

The state diagram of the Markov model used for recovery block dependability modeling is presented in Fig. 1. The states of this continuous time Markov chain for $1 \leq i \leq n$ alternates are defined as:

EX_i	–	execution of i-th alternate;
N_i	–	occurrence of a detected failure;
Z_i	–	occurrence of an undetected failure;
REC	–	execution of the global recovery;
F	–	failure state;
END	–	successful execution state.

We assume that the initial state is EX_1.

If there is no alternate or the acceptance test failure the correct result passes the acceptance test i.e. the recovery block system passes to the successful execution state END with transition rate μ_A. Similar failures in an alternate and the acceptance test cause an acceptance of the incorrect result i.e. the occurrence of undetected failure. In that case the system

passes to Z_i state with failure rate $c\lambda$. The occurrence of distinct failures in an alternate and the acceptance test enables the system to detect the failure. It means that the system passes from EX_i state to N_i state with transition rate $(1-c)(1+p)\lambda$. In that case the recovery is performed by executing the next alternate (if there is one) or by activating the global recovery. The system passes from the global recovery state REC to the successful execution state END with transition rate $r\mu_R$ or to the failure state F with transition rate $(1-r)\mu_R$, where r is the probability of successful global recovery.

It is clear that for $0 \leq r < 1$ the failure probability $P_F(t)$ is equal to the unreliability (obtaining no result or an undetected erroneous result). On the other hand, for $r = 1$ failure probability $P_F(t)$ is equal to the unsafety of the system (obtaining an undetected erroneous result). It does not matter for the safety of the system if there is no result, because in many cases there is enough time to invoke a higher level recovery procedure or to shut down the system.

In the study of Markov chains with absorbing states (F and END) the steady – state analysis is trivial and uninteresting, while transient analysis is of interest. The solution of the differential equations, to obtain time dependent state probabilities, is a formidable task in general [19]. One method of solving such differential equations is to use the Laplace transform, which simplifies the obtained system of differential equations to a system of algebraic equations. Taking Laplace transforms on the established differential equations we get:

$$P_{Z_i}(s) = \frac{[\mu_A(1-c)(1+p)\lambda]^{i-1}c\lambda}{[(s+\alpha_1)(s+\alpha_2)]^i}, \qquad 1 \leq i \leq n$$

$$P_{REC}(s) = \frac{[\mu_A(1-c)(1+p)\lambda]^n}{[(s+\alpha_1)(s+\alpha_2)]^n} \frac{1}{s+\alpha_3}$$

$$P_F(s) = \frac{1}{s}\left[\mu_A \sum_{i=1}^{n} P_{Z_i}(s) + (1-r)\mu_R\, P_{REC}(s)\right]$$

where

$$\begin{aligned} \alpha_1 &= \mu_A \\ \alpha_2 &= \mu_A + \lambda[1+p(1-c)] \\ \alpha_3 &= \mu_R. \end{aligned}$$

The rational function

$$F(s) = \left[\mu_A \sum_{i=1}^{n} P_{Z_i}(s) + (1-r)\mu_R\, P_{REC}(s)\right]$$

is expressed as the following sum

$$F(s) = \sum_{i=1}^{2}\sum_{j=1}^{n} \frac{A_{ij}}{(s+\alpha_i)^{n-j+1}} + \frac{A_{31}}{s+\alpha_3}.$$

which implies to inversion

$$f(t) = \sum_{i=1}^{2}\sum_{j=1}^{n} A_{ij}\, \frac{t^{n-j}}{(n-j)!}\, e^{-\alpha_i t}\, \delta(t) + A_{31}\, e^{-\alpha_3 t}\, \delta(t)$$

where

$$\delta(t) = \begin{cases} 1, & t \geq 0 \\ 0, & t < 0. \end{cases}$$

Finally, the failure probability of recovery block system is estimated by using integration:

$$P_F(t) = \int_0^t f(x)\, dx.$$

The product – form solutions for $P_F(t)$ for small n are presented in [4]. There is no convenient product – form solution for general case of n alternates. Therefore, in the following, we discuss the explicit matrix solution [19]. Let $P(t)$ be the row vector of the state probabilities for all states at time t. Now we define the transition rate matrix $A = [a_{ij}]$ so that the off – diagonal elements $a_{ij}(i \neq j)$ are positive transition rates, while the on – diagonal elements are the negative sum of the off – diagonal elements in the same row. The state probability vector is related to the transition rate matrix by the forward Kolmogorov equation in matrix form

$$\frac{dP(t)}{dt} = P(t)\, A$$

with initial condition $P(0) = [1, 0, \ldots, 0]$.

An obvious solution is given by $P(t) = P(0)\, e^{At}$, where the matrix function e^{At} has the the power series expansion:

$$e^{At} = \sum_{k=0}^{\infty} \frac{(At)^k}{k!}.$$

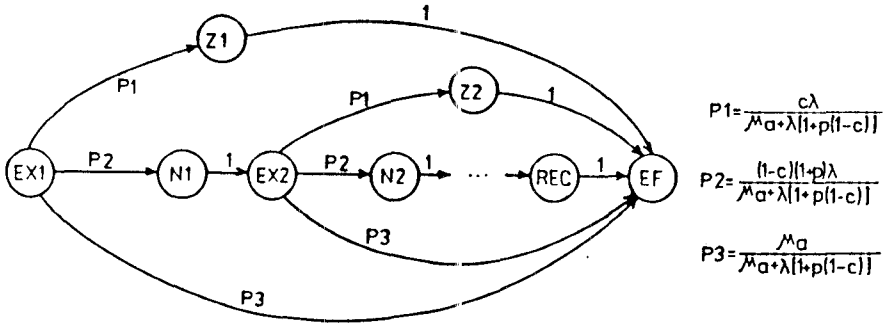

Figure 2: Embedded discrete time Markov chain

However, we are interested only in a failure probability:

$$P_F(t) = \sum_{k=0}^{\infty} \frac{(d_k t)^k}{k!}$$

where d_k is the element $a_{1(3n+2)}$ of the matrix A^k.

3.2 Average execution time modeling

In order to estimate the average execution time of the recovery block system we use the embedded discrete time Markov chain of the Markov reliability model. We have lumped the states END and F in one state EF which indicates the end of execution, as it can be seen in Fig. 2. EF is the absorbing state and the remaining $(3n+1)$ states are transient.

The transition probability matrix of this chain is partitioned so that

$$P = \left[\begin{array}{c|c} Q & C \\ \hline 0 & 1 \end{array}\right]$$

where Q is an $(3n+1)$ by $(3n+1)$ substochastic matrix describing the probabilities of transitions only among the transient states.

It can be shown that the inverse matrix $M = (I-Q)^{-1}$, so called fundamental matrix, exists. The fundamental matrix M is a rich source of information on the Markov chain. Element m_{ij} of matrix M denotes the average number of times the state S_j is visited before entering the absorbing state, given that the starting state is S_i [19]. Hence, the elements of the first row denote the average number of times the RB alternates are executed (i.e. states EX_i are visited) if the starting state is EX_1. When these values are multiplied with the average execution time of each state (average execution times of states N_i and Z_i are zero) for the average RB execution time we have obtained:

$$T = \frac{1 + p_2 + \cdots + p_2^{n-1}}{\mu_A} + \frac{p_2^n}{\mu_R}$$

and in that case we make the further definition

$$p_2 = \frac{(1-c)(1+p)\lambda}{\mu_A + \lambda[1+p(1-c)]}$$

which gives the conditional probability of going to state N_i on the next step, given that we are currently at the state EX_i.

4 Discussion of the main results

The presented model allows quantitative and probabilistic measurement of the effectiveness of a recovery block. It should be emphasized that the intention is to show the approximate behavior of the RB rather than derive any definite quantitative conclusions. As expected [15], reliability and safety reduce with the failure rate λ and execution time t. Furthermore, the failure probabilities of the RB system are very sensitive to the ratio $\lambda/\mu_A < 1$. Thus a general system reliability result is confirmed: the use of more reliable components leads to more reliable fault tolerant system.

Fig. 3 presents the unsafety (undetected failure probability) of the RB system with two alternatives. The increase of c causes the safety reduction, which

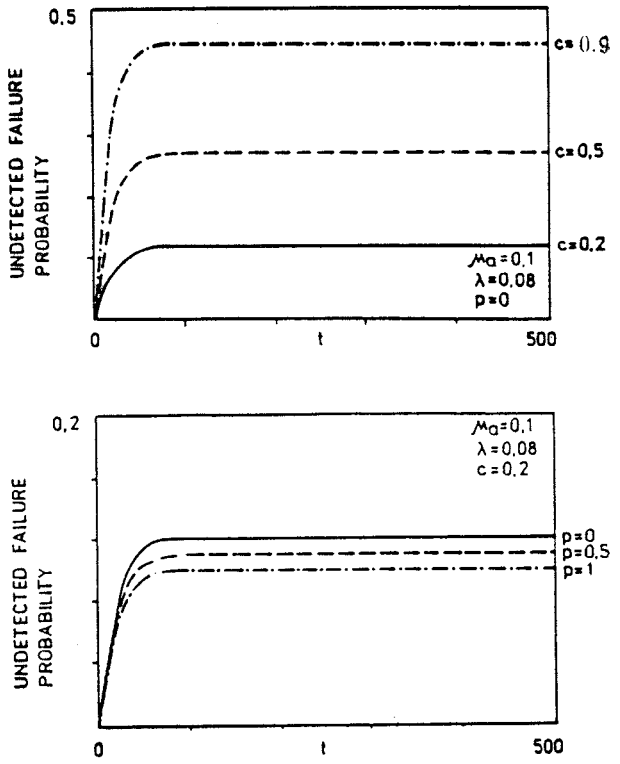

Figure 3: Undetected failure probability (unsafety)

confirms the results obtained in [2], [13]. On the contrary, for greater values of parameter p (more distinct acceptance test failures) the safety is insignificantly higher at the cost of lower reliability. In fact, p does not have any major affect on the safety. It means that only the coincident and similar failures between alternates and the acceptance test have significant impact on safety, which agrees with [2]. The probability of undetected failure cannot be eliminated by increasing the number of alternates (even for small values of c). Thus, considering safety, there is no need of using more than two alternates. On the contrary, it is better to use two than three (or more) alternates, when emphasis is put on safety rather than on reliability.

The total failure probability (the unreliability) of the RB system with two alternatives is presented in Fig. 4. In general, when the probability of global recovery r is increased the reliability also increases, and the influence of the number of alternates decreases.

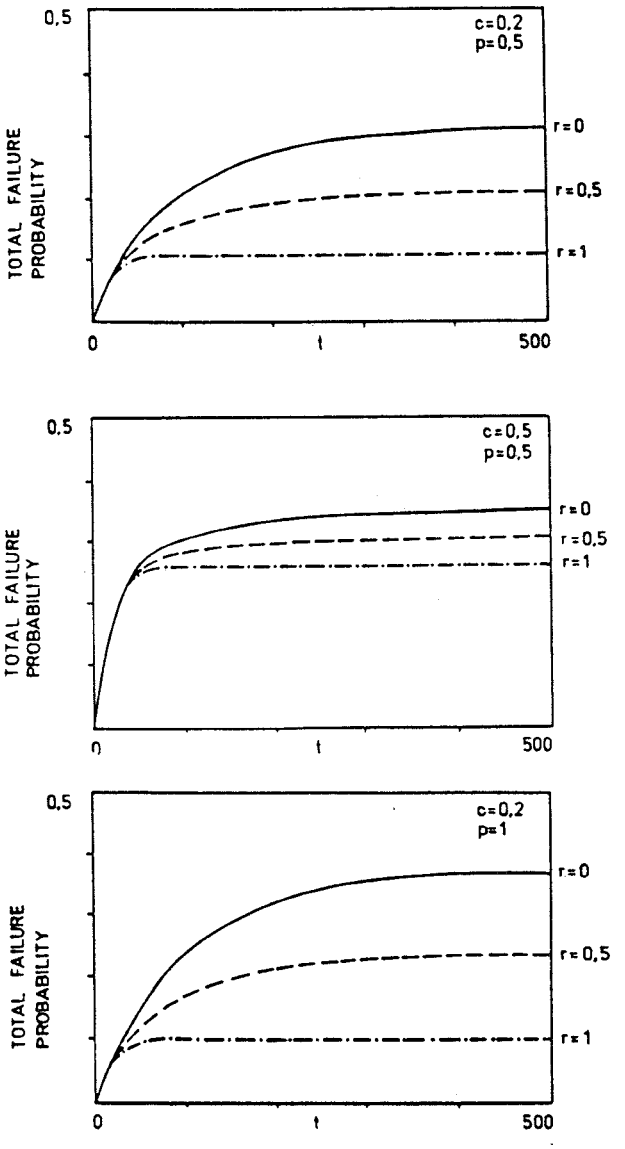

Figure 4: Total failure probability (unreliability) $\mu_A = 0.1$ and $\lambda = 0.08$

The occurrence of the coincident and similar failures between the alternates and the acceptance test reduces the reliability, as well as safety. In the case of greater c the influence of r on the reliability is smaller. As it can be seen, for greater values of the parameter p we obtain smaller reliability (unlike safety), and the influence of the global recovery and number of alternates is greater. Further on, the RB reliability (unlike safety) is improved by increasing the number of alternates, especially for small c. This again demonstrates the tradeoff between reliability and safety. However,

the reliability improvement is not proportional to the number of alternate blocks provided, which agrees with [3]. In addition, no reliability improvement has to be expected when using more than two alternates if coincident and similar failures between alternates and the acceptance test dominate. However, RB can be used to improve the reliability of software system. The amount of improvement significantly depends on the amount of coincident and similar failures between alternates and the acceptance test.

We have analyzed the influence of various system parameters on the average recovery block execution time (Fig. 5). As it was expected, average RB execution time increases as a function of the average alternate execution time $T = 1/\mu_A$. The average RB execution time (as the reliability and safety) is very sensitive to the variation of the failure rate λ, since the probability of the next alternate initialization depends on the failure rate. We get lower average RB execution time for higher values of parameter c (the occurrence of the undetected failure is more probable than the activation of the next alternate). The average RB execution time increases are obtained for greater values of the parameter p, because the greater probability of the next alternate activation is obtained for higher number of distinct acceptance test failures.

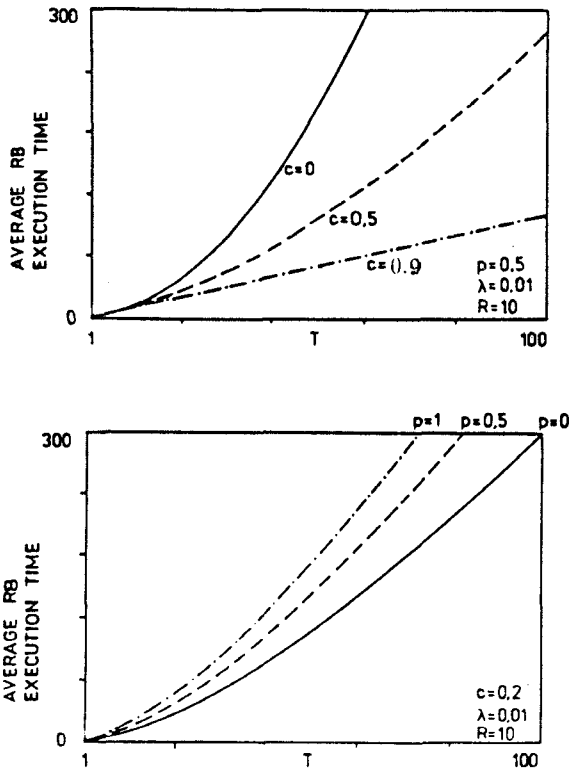

Figure 5: Average RB execution time

5 Conclusion

Fault tolerant systems are often used in computer controlled safety critical applications. For complex fault tolerant systems dependability modeling has become an integral part of the system design process. Thus, early analyses during system development are possible. This paper presents dependability model of the recovery block system during the period of execution. Unlike previous works, we carry out the analysis in the time domain. The model combines components information obtained from the software reliability models with information about the particular fault tolerant structure, possible types of failures and the ability to recover from failures. We use the continuous time Markov model for the general case of n alternates. In order to obtain the time dependent failure probabilities (i.e. reliability and safety) the transient analysis of the Markov chain is of interest. In this paper we derive the undetected failure probability (unsafety) and total failure probability (unreliability), as well as the average execution time of the recovery block system. Based on the theoretical analysis, the influence of the alternates and acceptance test quality on the reliability, safety and the average recovery block execution time is discussed. Both reliability and safety are fundamental qualities of real time systems, so their tradeoff is also discussed.

The presented model may constitute a framework for conducting experiment. Owing to the prominent influence of the acceptance test in the failure process of the RB, such an experiment should not be limited to examining the intra alternate correlations, but should cover the correlations between the alternate and the acceptance, as well.

References

[1] T.Anderson and P.A.Lee, *FAULT TOLERANCE - Principles and Practice*, Prentice/Hall International, 1981.

[2] J.Arlat, K.Kanoun and J.C.Laprie, "Dependability Modeling and Evaluation of Software Fault - Tolerant Systems", *IEEE Transaction on Computers*, Vol. 39, No. 4, pp. 504 - 513, April 1990.

[3] S.D.Cha,"A Recovery Block Model and Its Analysis", *Proceedings of the Fifth IFAC Workshop SAFECOMP'86*, Sarlat, France, pp. 21 - 26, 1986.

[4] K.Goševa - Popstojanova, *Software Fault Tolerant Reliability and Performance Modeling*, MSc Thesis, Faculty of Electrical Engineering, University "Kiril i Metodij", Macedonia, April 1990.

[5] K.Goševa - Popstojanova and A.Grnarov, "A New Markov Model of N Version Programming", *Proceedings of IEEE International Symposium on Software Reliability Engineering ISSRE'91*, Austin, Texas, USA, pp. 210 - 215, 1991.

[6] K.Goševa - Popstojanova and A.Grnarov, "Reliability Modeling and Evaluation of N Version Systems", *Preprints of SAFEPROCESS' 91, IFAC/IMACS Symposium on Fault Detection, Supervision and Safety for Technical Processes*, Baden-Baden, pp. 61 - 66, 1991.

[7] A.Grnarov, J.Arlat and A.Avizienis, "Modeling and Performance Evaluation of Software Fault Tolerance Strategies", *Proceedings of FTCS10*, Kyoto, Japan, pp. 251 - 253, 1980.

[8] J.Kelly, T.McVittie, W.Yamamoto, "Implementing Design Diversity to Achieve Fault Tolerance", *IEEE Software*, pp. 61 - 71, July 1991.

[9] J.Laprie, "Dependability Evaluation of Software Systems in Operation", *IEEE Transaction on Software Engineering*, Vol. SE-10, No.6, pp. 701 - 714, November 1984.

[10] J.Laprie, J.Arlat, C.Beounes, K.Kanoun, "Definition and Analysis of Hardware and Software Fault Tolerant Architectures", *IEEE Computer*, Vol.23, No.7, pp. 39 - 51, July 1990.

[11] J.Laprie et al. "The KAT (Knowledge - Action - Transformation) Approach to the Modeling and Evaluation of Reliability and Availability Growth", *IEEE Transaction on Software Engineering*, Vol.17, No.4, pp. 370 - 382, April 1991.

[12] B.Litlewood, "Software Reliability Model for Modular Program Structure", *IEEE Transactions on Reliability*, Vol. R-28, No.3, pp. 241 - 246, August 1979.

[13] M.Mulazzani, "Reliability Versus Safety", *Proceedings of the Fourth IFAC Workshop SAFECOMP'85*, Como, Italy, pp. 141 - 146, 1985.

[14] M.Mulazzani, K.Trivedi, "Dependability Prediction:Comparison of Tools and Techniques", *Proceedings of the Fifth IFAC Workshop SAFECOMP'86*, Sarlat, France, pp. 171 - 178, 1986.

[15] J.D.Musa, A.Iannino and K.Okumoto, *Software Reliability - Measurement, Prediction, Application*, Mc Grow-Hill, 1987.

[16] R.Scott, J.Gault and D.McAllister, "Fault Tolerant Software Reliability Modeling", *IEEE Transactions of Software Engineering*, Vol. SE-13, No. 5, pp. 582 - 592, May 1987.

[17] M.Shooman, "A Micro Software Reliability Model for Prediction and Test Apportionment", *Proceedings of ISSRE'91*, Austin, Texas, USA, pp. 52 - 59, 1991.

[18] Ed. S.K.Shrivastava, *Reliable Computer System*, Collected Papers of the Newcastle Reliability Project, Springer - Verlag Berlin Heidelberg, 1985.

[19] K.S.Trivedi, *Probability and Statistics with Reliability, Queuing and Computer Science Applications*, London: Prentice Hall Inc, 1982.