International Journal of Reliability, Quality and Safety Engineering
Vol. 7, No. 2 (2000) 97–111

TIMING ANALYSIS OF A FAULT-TOLERANT TECHNIQUE SUBJECT TO HARDWARE FAILURES

SERGEY V. GUROV, LEV V. UTKIN and IGOR B. SHUBINSKY
Department of Computer Science, Forest Technical Academy
Institutski per.5, St.Petersburg, 194021, Russia

Received 1 July 1996
Revised 15 February 2000

In this paper, we focus on the timing analysis of a fault-tolerance technique when a program is written in the form of several modules. If a hardware failure does occur during the ith module execution, the program must roll back and after restoring, the ith module execution is repeated. All random variables, such as time to hardware failure, replication time, and execution time of modules are arbitrarily distributed. We propose the efficient recurrence algorithms for computing the probability distribution and the mean value of execution time. The asymptotic behavior of the fault-tolerance technique is discussed. We prove that by increasing the number of modules, the mean execution time of a module tends to a defined value. The numerical examples are provided to illustrate and analyze the results. The application examples show that for the large variety of systems there exists an optimal number of modules minimizing the execution time.

Keywords: Software Fault Tolerance; Execution Time; Fuzzy Time; Recurrence Algorithm.

1. Introduction

There exist various techniques to protect a computer system from the consequences of hardware and software faults.[1] Let us consider one of them. Suppose that a program is written in the form of n modules. The output of each module is stored for its possible recursive usage in future. If a hardware failure does occur during the ith module execution, the program must roll back and after restoring, its ith module execution is repeated. Here we assume that only hardware failures can occur and detection of a failure is simultaneous. This fault-tolerance technique is well known and several authors have analyzed its reliability behavior.[2] It may be used in various types of computer systems: communications, database systems, real-time process-control systems. However, the most of the considered models describing the fault-tolerant techniques assume that the random time to failure and execution time are governed by the exponential distribution. This condition restricts their actual application. Therefore, in this paper we analyze the fault-tolerance technique under condition that all random variables, such as time to hardware failure, time to repair, and execution time (ET) of modules are arbitrarily distributed. Moreover, ETs of modules depend on initial data and generally may be unknown. In this case, we consider a fuzzy description[3] of ETs. Thus, the reliability behavior of the software-hardware system is defined by the following factors:

(1) The number of modules in the program.
(2) ETs of modules as random or fuzzy variables.
(3) Failure mode of the hardware.
(4) Time to repair as random variable.

An application of fault-tolerance techniques can cause an increase of the ET. Note that the ET is an important consideration for real-time systems because the reliability of such the systems depends on their ability to meet the critical task deadlines.[4] Moreover, there exist optimal values of n for some systems which minimize the mean execution time (MET) and these optimal values can be obtained by means of analyzing the ET. The ET is a random variable. Therefore, we study the basic probability characteristics of the ET. In order to investigate the behavior of the fault-tolerance technique by a large number of modules we analyze the asymptotic properties of the MET. Thus, in the paper we consider the following measures of system performance:

(1) Probability distribution of execution time.
(2) MET.
(3) Fuzzy MET.
(4) Asymptotic MET of one module.

The purposes of this paper are to propose efficient algorithms for computing above measures under general assumptions and to show how to reduce the MET by means of the optimal choice of the value n.

In Sec. 2 we consider the state transition and time diagrams of the analyzed system. The methods for computing the probability distribution and the mean value of the ET are provided in Sec. 3. In Sec. 4 we analyze a case when $n \to \infty$. The MET under condition of the fuzzy ETs of modules is studied in Sec. 5. The applications of proposed methods to communications, database systems, and real-time process-control systems are considered in Sec. 6. Appendix contains the proof of the basic results. We will denote

pdf, *Cdf*, sf probability density, distribution, survivor functions, respectively;

$f(t), F(t), \overline{F}(t)$, T_0 *pdf*, *Cdf*, sf, expectation of time to hardware failure, respectively;

$g(t), G(t), \overline{G}(t)$, T_r *pdf*, *Cdf*, sf, expectation of time to repair, respectively;

$h_i(t), H_i(t), \overline{H}_i(t), T_i$ *pdf*, *Cdf*, sf, expectation of the ith module ET, respectively;

$\mu_A(x)$ membership function of the fuzzy set $\widetilde{A}$;

$f_s(t) = f(t+s)$;

$*$ convolution symbol $f * g(t) = \int_0^t f(u)g(t-u)du$;

$f * 1(t) = \int_0^t f(u)du$;

$f^{*(k)}(t)$ k-fold convolution of the function $f(t)$.

2. The State Transition and Time Diagrams

Suppose a program consists of n modules, i.e., there are n stages of its execution. The output data of the ith module are used as the input data of the $(i+1)$th module. We assume that ETs of all the modules are independent random variables with the pdf $h_i(t)$, $i = 1, \ldots, n$. The hardware may fail and repair. Let the time to hardware failure and time to hardware repair be random variables having pdf $f(t)$ and $g(t)$, respectively.

A state transition diagram of the system is given in Fig. 1. The system can be in one of the following states:

k, $k = 1, 2, \ldots, n$: hardware is in operating state and kth module of the program is executed;

$\overline{k}$, $k = 1, 2, \ldots, n$: hardware is under repair and the program is stopped (in idle state) during the kth module execution;

$n+1$: the program completes.

An example of a time diagram of the system is given in Fig. 2, where the following time instances are depicted: (1) first module completes; (2) first hardware failure occurs; (3) repair completes and second module is repeated; (4) second hardware failure occurs; (5) repair completes and second module is repeated; (6) second module completes.

Let $p_k(t)$ and $p_{\overline{k}}(t)$ be the probabilities that the system is in states k and $\overline{k}$ at time t, respectively. The probability that the program completes during time interval $[0, t]$ is defined as

$$F(t) = 1 - \sum_{k=1}^{n} \left(p_k(t) + p_{\overline{k}}(t)\right) = p_{n+1}(t). \tag{1}$$

Let T_k, $T_{\overline{k}}$ be the mean sojourn times in kth and $\overline{k}$th states, respectively, and T be the MET of the program. Then

$$T_k = \int_0^\infty p_k(t)dt, \; T_{\overline{k}} = \int_0^\infty p_{\overline{k}}(t)dt, \; T = \sum_{k=1}^{n}(T_k + T_{\overline{k}}). \tag{2}$$

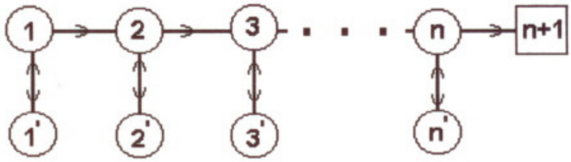

Fig. 1. The state transition diagram of the system.

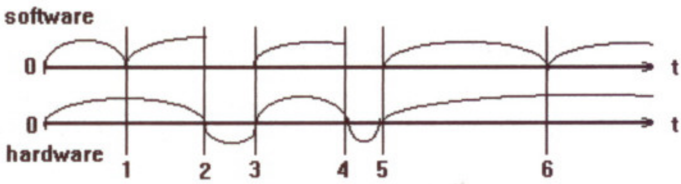

Fig. 2. An example of the time diagram.

3. Reliability Measures

In this section we consider the probability distribution of the ET and MET.

Result 1. *Denote*

$$\overline{\omega}_k(t) = \sum_{j=0}^{\infty} \left(f\overline{H}_k\right)^{*(j)} * g^{*(j)}(t). \tag{3}$$

Then the state probabilities of the system can be computed as

$$p_k(t) = \sum_{i=1}^{k} a_i * (\overline{F}h_i * \cdots * h_{k-1} * \overline{H}_k)(t), \qquad k = 1, \ldots, n, \tag{4}$$

$$p_{\overline{k}}(t) = \overline{G} \sum_{i=1}^{k} a_i * (fh_i * \cdots * h_{k-1} * \overline{H}_k)(t), \qquad k = 1, \ldots, n, \tag{5}$$

$$p_{n+1}(t) = \sum_{i=1}^{k} a_i * (\overline{F}h_i * \cdots * h_n) * 1(t). \tag{6}$$

Here the function $a_k(t)$ is recurrently determined from $a_1(t) = \overline{\omega}_k(t)$ and

$$a_k(t) = \overline{\omega}_k * g * \sum_{i=1}^{k-1} a_i(fh_i * \cdots * h_{k-1} * \overline{H}_k)(t). \tag{7}$$

The proof of Result 1 can be found in the Appendix. By using Eqs. (1) and (3)–(7), we obtain the probability distribution of the ET. The main difficulty in computation of $F(t)$ is expression (3) because of the infinite number of terms. However, the time to hardware failure is much more than the duration of one stage in various cases. Therefore, $(f\overline{H}_k)^{*(j)} \to 0$ as $j \to \infty$. This implies that for approximate computing (3) we should calculate only the some terms.

Now consider the MET T. It can be computed by means of a simple recurrence algorithm.

Result 2. *The MET T is computed as follows:*

$$T = \sum_{i=1}^{n} \overline{a}_i \int_0^\infty (\overline{F} + T_r f)(t) \overline{H}^{(i)}(t) dt\,, \tag{8}$$

where

$$\overline{H}^{(i)}(t) = 1 - \int_0^t h_i * \cdots * h_n(\tau) d\tau\,,$$

values $\overline{a}_k$, $k = 1, 2, \ldots, n$, are recurrently determined from expressions

$$\overline{a}_1 = \frac{1}{1 - \displaystyle\int_0^\infty f(t) \overline{H}_1(t) dt}\,,$$

$$\overline{a}_k = \frac{\displaystyle\sum_{i=1}^{k-1} \overline{a}_i \int_0^\infty f(t) h_i * \cdots * h_{k-1} * \overline{H}_k(t) dt}{1 - \displaystyle\int_0^\infty f(t) \overline{H}_k(t) dt}\,. \tag{9}$$

The proof of Result 2 can be found in the Appendix. From Eq. (8), we can see that the value of T is independent of the distribution of time to repair and is determined only by the expectation T_r.

Let us consider a special case when $h_k(t) = \delta_{x_k}(t)$, where $\delta_{x_k}(t)$ is the standard impulse function: it has unit area concentrated in the immediate vicinity of $t = x_k$. Then there holds

$$h_i * \cdots * h_{k-1} \overline{H}_k(t) = \begin{cases} 1\,, & \text{if } t \in [x_i + \cdots + x_{k-1},\, x_i + \cdots + x_{k-1} + x_k]\,, \\ 0\,, & \text{otherwise}\,. \end{cases}$$

Therefore, it follows from Eq. (9) that

$$\overline{a}_1 = \frac{1}{\overline{F}(x_1)}\,,$$

$$\overline{a}_k = \sum_{i=1}^{k-1} \overline{a}_i \frac{\overline{F}(x_i + \cdots + x_{k-1}) - \overline{F}(x_i + \cdots + x_{k-1} + x_k)}{\overline{F}(x_k)}\,.$$

Denoting $\Phi(t) = \int_0^\infty \overline{F}(x + t) dx$ and using Eq. (8), we obtain

$$T = \sum_{i=1}^{n} \overline{a}_i (T_0 + T_r - (\Phi + T_r \overline{F})(x_i + \cdots + x_n))\,. \tag{10}$$

Thus, we have obtained extremely simple recurrence expressions for computing T.

4. An Asymptotic Property

Let us consider the following ratio:

$$\bar{t}_n = \frac{\text{mean execution time}}{\text{number of modules}} = \frac{T}{n}\,.$$

We can show that under some conditions, there exists the following limit:

$$\bar{t} = \lim_{n\to\infty} \bar{t}_n\,.$$

Result 3. *If ETs of all the modules are independent random variables with the identical pdf $h(t)$ and with expectations $\bar{t}$, then there holds*

$$\bar{t} = \frac{T_0 + T_r}{\displaystyle\int_0^\infty \sum_{k=1}^{\infty} h^{*(k)}(x)\overline{F}(x)dx}\,.$$

Result 3 implies that by increasing the number of modules, the MET of each module is constant and cannot be decreased.

Example. Let us determine how the ratio $\bar{t}_n$ depends on the value n under the following conditions:

— number of modules $n = 1 \div 200$;
— execution time of each module is equal to 5;
— time to system failure has the gamma distribution with the scale parameter $\lambda = 0.05$ and fixed shape parameters $k = 5$ and 10;
— time to repair is arbitrarily distributed with the expectation $T_r = 1$.

Figure 3 illustrates the functions $\bar{t}_n$ versus n for different values of k. Note that the function $\bar{t}_n$ tends to the steady-state value $\bar{t}$. If $k = 5$, then $\bar{t} = 5.179$. If $k = 10$, then $\bar{t} = 5.088$.

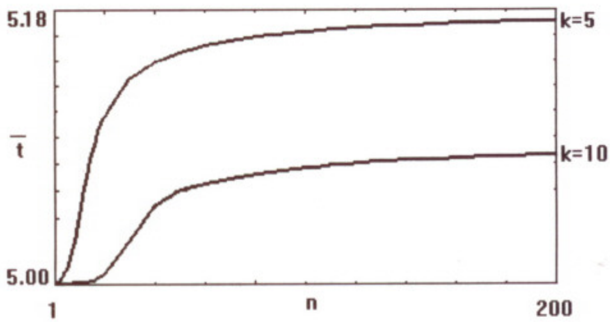

Fig. 3. The illustration of the steady-state behavior of $\bar{t}_n$.

5. Fuzzy Execution Time

In previous sections, we assumed that ETs of modules are random or deterministic. Now we consider a case when ETs of modules are fuzzy numbers with the membership functions $\mu_i(x)$, $i = 1, \ldots, n$. Obviously, in this case the MET of the program is a fuzzy number $\widetilde{T}$. In accordance with Zadeh's extension principle[3], we can write the membership function of the fuzzy MET as follows:

$$\mu_T(u) = \sup_{x_1 \geq 0, \ldots, x_n \geq 0} \left\{ \min_{i=1,\ldots,n} \mu_i(x_i) | T(x_1, x_2, \ldots, x_n) = u \right\}. \tag{11}$$

Here $T(x_1, x_2, \ldots, x_n)$ is a value of the MET under condition that ETs of modules are deterministic and equal $x_1, x_2, \ldots, x_n$, respectively. Thus, the values $T(x_1, x_2, \ldots, x_n)$ are determined by means of Eq. (10). It should be noted that computation of $\mu_T(u)$ by means of Eq. (11) is an extremely difficult problem. Therefore, we need another way to compute $\mu_T(u)$.

Note that $T(x_1, x_2, \ldots, x_n)$ is the nondecreasing function of variables x_i, $i = 1, \ldots, n$. This implies that we can use a method of α-cut intervals[5] for computing $\mu_T(u)$. Denote

$$x_{i\alpha}^- = \inf\{x | \mu_i(x) \geq \alpha\}, \qquad x_{i\alpha}^+ = \sup\{x \mid \mu_i(x) \geq \alpha\}, \qquad i = 1, \ldots, n,$$
$$T_\alpha^- = \inf\{u | \mu_T(u) \geq \alpha\}, \qquad T_\alpha^+ = \sup\{u | \mu_T(u) \geq \alpha\}, \qquad \alpha \in [0, 1].$$

According to the method of α-cut intervals, we can write

$$T_\alpha^- = T(x_{1\alpha}^-, \ldots, x_{n\alpha}^-),\ T_\alpha^+ = T(x_{1\alpha}^+, \ldots, x_{n\alpha}^+). \tag{12}$$

Hence

$$\mu_T(T_\alpha^-) = \alpha,\ \mu_T(T_\alpha^+) = \alpha.$$

Further application of the fuzzy ET depends on a specific problem. If we compare two systems, we can use algorithms for ranking fuzzy numbers based on ranking indices.[5] One of the most accurate in results and efficient in computation algorithms is presented by Tseng and Klein.[6] If we have to estimate one system, then a crisp value of the fuzzy ET can be found by using algorithms of the defuzzification. Two of the more common techniques of the defuzzification are the "centroid" and "maximum" methods. In the centroid method, the crisp value of a fuzzy number is computed by finding the value of the center of gravity of the membership function. In the maximum method, one of the values at which the membership function of the fuzzy number has its maximum value is chosen as the crisp value. To choose a system having a higher degree of satisfying a real-time constraint, we must compare fuzzy execution time with a crisp value of the constraint. This is a special case of comparing the fuzzy numbers when one number is nonfuzzy. Then, the value of an ranking index can be considered as a degree of satisfying the real-time constraint.

6. Application Examples

Example 1. Let us consider a communication system as a component of a telephone network. The transmission speed of the system equals 9600 bit/second. A transmitted message has the size 351.56 kb. Suppose that the channel is ideal without the transmission link and switch failures. Then the required transmission time for the message equals 300 s. The message can be transmitted as a single block and can be split into n blocks which are consecutively transmitted as separate messages. However, due to a control token and a self-synchronizing code, the transmission time of each block increases on Δt (s). If a failure of the system occurs during transmission, then the currently transmitted block is sent again. The time to system failure has the gamma distribution with the scale parameter $\lambda = 0.05\ \text{s}^{-1}$ and the fixed shape parameter $k = 5$. The mean time to restore the communication process equals 1 s. Figure 4 shows curves of the mean transmission time T versus the number of blocks n for various values of $\Delta t(\Delta t = 0.05, \Delta t = 0.1, \Delta t = 0.2)$. We can see that all the curves in Fig. 4 have a minimum point. Therefore, there exists an optimal value n_{opt} minimizing the mean transmission time. If we choose $n = n_{\text{opt}}$, we obtain desired results (see Table 1). From Table 1, we conclude that the small variation of the control token and self-synchronizing code sizes may cause significant changes in the structure of the message.

Example 2. To illustrate the resultant fuzzy MET we consider two real-time process-control systems. The programs of first and second systems consist of $n_1 = 10$ and $n_2 = 5$ subprograms, respectively. The time to hardware failure has the gamma distribution with the scale parameter $\lambda = 0.001\ \text{s}^{-1}$ and the fixed shape

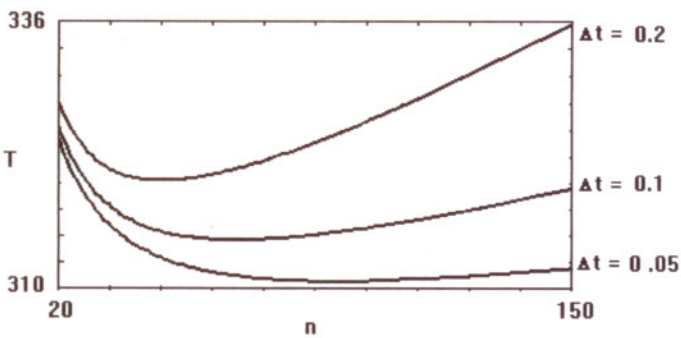

Fig. 4. The mean transmission time versus n for different Δt.

Table 1. The values of n_{opt} and T versus Δt.

Δt, (s)	n_{opt}	T, (s)
0.05	90	311.7
0.1	64	315.6
0.2	46	321.3

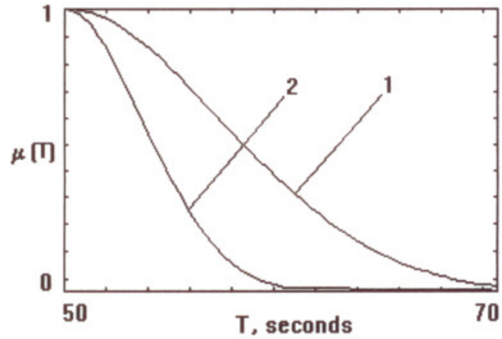

Fig. 5. The membership functions of the fuzzy execution time.

Table 2. The values of n_{opt} and T versus λ.

λ, s^{-1}	n_{opt}	T, s
0.001	1	27.2728
0.005	18	27.2934
0.01	35	27.3286

parameter $k = 2$, the time to repair is arbitrarily distributed with the expectation $T_r = 10$ s. Let $\mu_i(x_i) = \exp(-(x-5)^2)$ and $\mu_i(x_i) = \exp(-(x-10)^2)$ be the membership functions of the fuzzy ith subprogram ETs for first and second systems, respectively. There is the real-time constraint of 56 s. We should choose the system which reduces the risk that the program will not be completed within the real-time constraint. Figure 5 shows the membership functions of the fuzzy ETs for system 1 (curve 1) and system 2 (curve 2). By using the index of Tseng and Klein[6], we obtain the degree of satisfying the real-time constraint 0.294 and 0.767 for the second system. This implies that the second system is more acceptable.

Example 3. Let us consider a database system. The database with the size 60 000 kb is stored on a hard disk QUANTUM TRB850A working on IBM PC with the data transfer rate 2200 kb/s. This database can be implemented as one or more files. Each file contains a descriptor word index with the size 0.8 kb. How many files n_{opt} should be created to minimize the mean time of reading the database under condition that the time to hardware failure has the gamma distribution with the scale parameter λ and the fixed shape parameter $k = 2$, the time to repair is arbitrarily distributed with the expectation $T_r = 1$ s? The computational results are shown in Table 2.

7. Conclusion

In this paper, we have proposed the method for analyzing the well-known fault-tolerant technique by arbitrarily distributed random time to hardware failure, time to hardware repair, and execution time of a program. As such the steady-state characteristic as the MET can be computed by means of the very simple algorithm. The simplicity of this algorithm allows us to analyze the fault-tolerant technique when ETs of modules are fuzzy numbers.

In addition, the following conclusion can be made:

- The MET is independent of the distribution of time to hardware repair and is determined only by its expectation.
- By increasing the number of modules, the mean execution time of a module tends to a steady-state value.
- For various system, there exists an optimal number of modules minimizing the MET.
- The fuzzy reliability of the system can be computed by means of the simple method of α-cut intervals.

As we have seen from the application examples the use of the proposed method concerning the software can be extended on the communication and database systems. It should be noted that the considered model of the fault-tolerant technique and the proposed method do not take into account the various features of the hardware–software systems. However, they can be regarded as a tool for the preliminary evaluation and development of the optimal fault-tolerant technique. At the same time, further study is needed to develop efficient methods for analyzing reliability of complex systems by more realistic assumptions.

Appendix: The Proof of Results

Lemma 1. *The following equality is valid*:

$$\int_0^t h(x)a * (gf_{x+s})(t-x)dx = a * (g * hf_s)(t)\,.$$

Proof. By using the definition of the convolution, we obtain

$$\begin{aligned}\int_0^t h(x)a * (gf_{x+s})(t-x)dx &= \int_0^t h(x)\int_0^{t-x} a(\alpha)g(t-x-\alpha)f_s(t-\alpha)d\alpha dx\\ &= \int_0^t a(\alpha)f_s(t-\alpha)d\alpha\int_0^{t-\alpha} h(x)g(t-x-\alpha)dx\\ &= a * (g * hf_s)(t)\,,\end{aligned}$$

as was to be proved. □

Theorem 1. *The equation*

$$y(s_0, s, t) = \int_0^t (f_{s_0}h_s) * g(x)\int_0^\infty y(0, s, t-x)dsdx + \varphi(s_0, s, t) \tag{A.1}$$

has the following solution

$$y(s_0, s, t) = (f_{s_0} h_s) * \overline{\omega} * g * \int_0^\infty \varphi(0, s, t) ds + \varphi(s_0, s, t), \tag{A.2}$$

where

$$\overline{\omega}(t) = \sum_{j=0}^{\infty} (f\overline{H})^{*(j)} * g^{*(j)}(t). \tag{A.3}$$

Proof. Let us write Eq. (A.1) in the following form:

$$y(s_0, s, t) = (f_{s_0} h_s) * g * z(t) + \varphi(s_0, s, t), \tag{A.4}$$

where $z(t) = \int_0^\infty y(0, s, t) ds$. Then, it follows from Eq. (A.4) that

$$z(t) = (f\overline{H}) * g * z(t) + \int_0^\infty \varphi(0, s, t) ds.$$

By using Eq. (A.3), we obtain

$$z(t) = \overline{\omega} * \int_0^\infty \varphi(0, s, t) ds.$$

By substituting the function $z(t)$ into Eq. (A.4), we arrive at Eq. (A.2). □

Proof of Result 1. Let $y_k(s_0, s, t)$ be the pdf that the system is in kth state at time t. Here s_0 is the residual time to failure, s is the residual kth module ET. Let $y_{\overline{k}}(\tau_0, t)$ be the pdf that the system is in $\overline{k}$th state at time t. Here τ_0 is the residual time to repair. Let $y_{n+1}(s_0, t)$ be the pdf that the system is in $(n+1)$ th state at time t. Here s_0 is the residual time to failure. These functions satisfy the following system of integral equations[7,8]:

$$\begin{cases} y_1(s_0, s, t) = \int_0^t f(x + s_0) h_1(x + s) y_{\overline{1}}(0, t - x) dx + f(t + s_0) h_1(t + s), \\ y_k(s_0, s, t) = \int_0^t f(x + s_0) h_k(x + s) y_{\overline{k}}(0, t - x) dx \\ \qquad + \int_0^t h_k(x + s) y_{k-1}(x + s_0, 0, t - x) dx, \qquad k = 2, 3, \ldots, n, \\ y_{\overline{k}}(\tau_0, t) = \int_0^t g(x + \tau_0) \int_0^\infty y_k(0, s, t - x) ds dx, \qquad k = 1, 2, \ldots, n, \\ y_{n+1}(s_0, t) = \int_0^t y_n(s_0, 0, t - x) dx. \end{cases} \tag{A.5}$$

Then, the state probabilities are computed as follows:

$$p_k(t) = \int_0^\infty \int_0^\infty y_k(s_0, s, t) ds_0 ds\,, \qquad k = 1, \ldots, n\,, \tag{A.6}$$

$$p_{\overline{k}}(t) = \int_0^\infty y_{\overline{k}}(\tau_0, t) d\tau_0\,, \qquad k = 1, \ldots, n\,, \tag{A.7}$$

$$p_{n+1}(t) = \int_0^\infty y_{n+1}(s_0, t) ds_0 ds\,. \tag{A.8}$$

By eliminating from Eq. (A.5) equations corresponding to failed states, and by using Theorem 1, we obtain after simplification

$$y_1(s_0, s, t) = (f_{s_0} h_{1,s}) * \overline{\omega}_1(t)\,, \tag{A.9}$$

$$y_k(s_0, s, t) = \int_0^t h_k(x+s) y_{k-1}(x+s_0, 0, t-x) dx$$
$$+ (f_{s_0} h_{k,s}) * \overline{\omega}_k * g * \int_0^t \overline{H}_k(x) y_{k-1}(x, 0, t-x) dx\,,$$
$$k = 2, 3, \ldots, n\,. \tag{A.10}$$

Expressions (A.9) and (A.10) are recurrence relations for computing functions $y_k(s_0, s, t)$, $k = 1, 2, \ldots, n$. Let us obtain a more simple form of expressions (A.9) and (A.10). Introduce functions

$$a_k(t) = \begin{cases} \overline{\omega}_1(t)\,, & \text{if } k = 1\,, \\ \overline{\omega}_k * g * \int_0^t \overline{H}_k(x) y_{k-1}(x, 0, t-x) dx\,, & \text{if } k = 2, \ldots, n\,. \end{cases} \tag{A.11}$$

Then, expressions (A.9) and (A.10) can be rewritten as follows:

$$y_1(s_0, s, t) = a_1 * (f_{s_0} h_{1,s})(t)\,, \tag{A.12}$$

$$y_k(s_0, s, t) = \int_0^t h_k(x+s) y_{k-1}(x+s_0, 0, t-x) dx + a_k * (f_{s_0} h_{k,s})(t)\,. \tag{A.13}$$

Hence an explicit form of Eqs. (A.12) and (A.13) is

$$y_k(s_0, s, t) = \sum_{i=1}^{k} a_i * (f_{s_0} h_i * \cdots * h_{k-1} * h_{k,s})(t)\,. \tag{A.14}$$

Indeed, if $k = 1$, then we have Eq. (A.12). Let Eq. (A.14) be valid for the function $y_{k-1}(s_0, s, t)$. Then from Eq. (A.13), we obtain

$$y_k(s_0, s, t) = \sum_{i=1}^{k-1} \int_0^t h_{k,s}(x) a_i * (f_{x+s_0} h_i * \cdots * h_{k-1})(t-x) dx + a_k * (f_{s_0} h_{k,s})(t)\,.$$

By using Lemma 1, we arrive at Eq. (A.14). Now by using Eq. (A.14), we can obtain recurrence relations for computing functions $a_k(t)$. It follows from Eq. (A.11) that

$$a_k(t) = \overline{\omega}_k * g * \sum_{i=1}^{k-1} \int_0^t \overline{H}_k(x) a_i * (f_x h_i * \cdots * h_{k-1})(t-x)dx\,.$$

By using Lemma 1, we arrive at Eq. (7). The functions $y_{\overline{k}}(\tau_0, t)$ and $y_{n+1}(s_0, t)$ can be written as follows:

$$y_{\overline{k}}(\tau_0, t) = g_{\tau_0} * \sum_{i=1}^{k} a_i * (f h_i * \cdots * h_{k-1} * \overline{H}_k)(t)\,,$$

$$y_{n+1}(s_0, t) = \sum_{i=1}^{k} a_i * (f_{s_0} h_i * \cdots * h_n) * 1(t)\,. \tag{A.15}$$

From Eqs. (A.6)–(A.8), (A.14), and (A.15), we obtain the state probabilities. □

Proof of Result 2. By integrating Eqs. (4) and (5) with respect to t from 0 to ∞, and denoting $\overline{a}_i = \int_0^\infty a_i(t)dt$, we obtain

$$T_k = \sum_{i=1}^{k} \overline{a}_i \int_0^\infty \overline{F}(t) h_i * \cdots * h_{k-1} * \overline{H}_k(t)dt\,,$$

$$T_{\overline{k}} = T_r \sum_{i=1}^{k} \overline{a}_i \int_0^\infty f(t) h_i * \cdots * h_{k-1} * \overline{H}_k(t)dt\,.$$

From Eq. (2), we obtain

$$T = \sum_{k=1}^{n} \sum_{i=1}^{k} \overline{a}_i \int_0^\infty (\overline{F} + T_r f)(t) h_i * \cdots * h_{k-1} * \overline{H}_k(t)dt\,.$$

From the following notation

$$\overline{H}^{(i)}(t) = \sum_{k=1}^{n} h_i * \cdots * h_{k-1} * \overline{H}_k(t) = 1 - \int_0^t h_i * \cdots * h_n(\tau)d\tau\,,$$

we arrive at Eq. (8). □

Proof of Result 3. For the steady-state behavior of the system, the state transition diagram can be reduced (see Fig. 6). We stick together states $k = 1, \ldots, n$ and obtain one state (0). Also we stick together states $\overline{k} = 1, \ldots, n$ and obtain another state (1). Note that at time $t \to \infty$ the value $\overline{t}$ is equal to the ratio of the time t to the mean number of transitions $M(t)$ from one module to another one during the time interval $[0, t]$, i.e.

$$\overline{t} = \lim_{t\to\infty} \frac{t}{M(t)} = \frac{1}{\omega}\,,$$

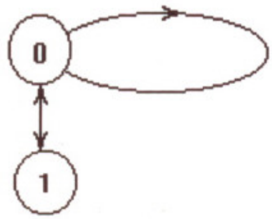

Fig. 6. The reduced state transition diagram.

where ω is the steady-state probability that at time $t \to \infty$ one module is executed and at time $t + dt$, $dt \to 0$ another module is executed. Let $y_0(s_0, s)$ be the pdf that the system is in state 0 at time $t \to \infty$. Here s_0 is the residual time to failure and s is the residual execution time. Let $y_1(\tau_0)$ be the pdf that the system is in state 1 at time $t \to \infty$. Here τ_0 is the residual time to repair. Then the following system of integral equations is valid:

$$\begin{cases} y_0(s_0, s) = \int_0^\infty f(x + s_0)h(x + s)y_1(0)dx + \int_0^\infty h(x + s)y_0(x + s_0, 0)dx\,, \\ y_1(\tau_0) = \int_0^\infty g(x + \tau_0) \int_0^\infty y_0(0, s)dsdx\,. \end{cases}$$

This system is a special case of system (A.5) as $t \to \infty$. Its solution is of the form

$$y_0(s_0, s) = \frac{1}{T_0 + T_r} \sum_{k=0}^{\infty} \int_0^\infty f_{s_0}(x) h^{*(k)} * h_s(x)dx\,,$$

$$y_1(\tau_0) = \frac{\overline{G}(\tau_0)}{T_0 + T_r}\,.$$

This implies

$$\omega = \int_0^\infty y_0(s_0, s)ds_0 = \frac{1}{T_0 + T_r} \sum_{k=1}^{\infty} \int_0^\infty \overline{F}(x) h^{*(k)}(x)dx\,,$$

as was to be proved. □

References

1. R. E. Gantenbein, T. F. Lawrence, and S. Y. Shin, "A framework for adaptive fault management in distributed computing systems," *International Journal of Reliability, Quality and Safety Engineering* **1** (1994), pp. 391–412.
2. R. V. Kakubava, R. K. Kukava, M. Sh. Kurtser, and I. S. Mikadze, "On feasibility of job execution with regard for computer reliability," *Automatics and Telemechanics* **7** (1981), pp. 176–181. (In Russian)
3. L. A. Zadeh, Fuzzy sets as a basis for a theory of possibility, *Fuzzy Sets Syst.* **1** (1978), pp. 3–28.

4. C. M. Krishna, Timing analysis of expedient voting and runahead, *Microelectronics Reliability* **34** (1994), pp. 765–770.
5. D. Dubois and H. Prade, *Possibility Theory: An Approach to Computerized Processing of Uncertainty* (Plenum Press, New York, 1988).
6. T. Y. Tseng and C. M. Klein, "New algorithm for the ranking procedure in fuzzy decisionmaking," *IEEE Trans. Syst. Man. and Cybernetics* **SMC-19** (1989), pp. 1289–1296.
7. S. V. Gurov and L. V. Utkin, "The time-dependent availability of repairable m-out-of-n and cold standby systems by arbitrary distributions and repair facilities," *Microelectronics Reliability* **35** (1995), pp. 1377–1393.
8. S. V. Gurov and L. V. Utkin, "Reliability of repairable systems with periodic modifications," *Microelectronics Reliability* **36** (1996), pp. 27–35.

About the Authors

Sergey V. Gurov received his M.Sc. (1968) from the State University in Leningrad. He obtained his Ph.D. in mathematics (1973) from the same university and the D.Sc. degree in control systems (1997) from the St. Petersburg Forest Technical Academy. Currently a professor in the Department of Computer Science, St. Petersburg Forest Technical Academy, his research interests include reliability theory, computational algorithms, and optimization.

Lev V. Utkin received his M.Sc. (1986) from the Institute of Electrical Engineering in Leningrad. He obtained his Ph.D. in information processing and control systems (1989) from the same institute. Currently an associate professor in the Department of Computer Science, St. Petersburg Forest Technical Academy, his research interests include fuzzy reliability theory, the theory of imprecise probabilities, imprecise probability models in the reliability theory, optimization, and uncertainty representation.

Igor B. Shubinsky received his M.Sc. (1961) from the Central Military University of Ukraine. He obtained his Ph.D. (1968) from the Moscow Institute of Automatic Machinery and the D.Sc. degree (1987) from the Glushkov Institute of Cybernetics (Kiev). Currently a Head of the Department of Computer Science, St. Petersburg Forest Technical Academy, his research interests include reliability theory, safety and fault-tolerant technique. He is an author of series books on fault-tolerant technique and computing.