Technical science and innovation

Volume 2021 | Issue 2 Article 3

6-26-2021

Assessing the Software Reliability of the Process Control Systems

Kh S. Turaev
Tashkent State Technical University named after Islam Karimov, 100095, Tashkent, Uzbekistan, mr.aaa_93@mail.ru

N A. Yusupbekov
Tashkent state technical university named after Islam Karimov, 100095, Tashkent, Uzbekistan

V B. Tarasov
Moscow State Technical University named after N.E. Bauman

Recommended Citation
Turaev, Kh S.; Yusupbekov, N A.; and Tarasov, V B. (2021) "Assessing the Software Reliability of the Process Control Systems," *Technical science and innovation*: Vol. 2021: Iss. 2, Article 3.
DOI: https://doi.org/10.51346/tstu-01.21.2-77-0119
Available at: https://btstu.researchcommons.org/journal/vol2021/iss2/3

13. Shinohara, K., Takaki, R. Contact finite element analysis swivel for large deployable space antenna on the satellite // IEEE Aerospace conference 2013. 2013. T. 20. S. 1-20. DOI: 10.1109/AERO.2013.6496896.
14. Jrusalem P. A., Poel, A. N. Numerical solution of the problem of deformation of a parabolic antenna // Science. Industry. Defense: tr. 17 Vseros. nauch. - tehn. conf., Novosibirsk, 20-22 Apr. 2016: in 4 vols. 1. Novosibirsk: Publishing House of NSTU, 2016. pp. 209-214.
15. Qiu D., Sun M., Wang Z., Wang Y., Chen Z. Practical Wind-Disturbance Rejection for Large Deep Space Observatory Antenna // IEEE Transactions on Control Systems Technology. 2014. Vol. 22. P. 1983–1990. DOI: 10.1109/TCST.2013.2296935.

UDC 681.5

ASSESSING THE SOFTWARE RELIABILITY OF THE PROCESS CONTROL SYSTEMS

Kh.S. Turaev[1*], A.N. Yusupbekov[1], V.B. Tarasov[2]

[1]Tashkent state technical university named after Islam Karimov,
Tashkent, Universitet 2, 100009
[2]"Moscow State Technical University named after NE Bauman (National Research University)"
(MSTU named after NE Bauman)

Abstract: *Methods of increasing the reliability of components and assemblies of control systems for technological processes and production, as well as methods of using fault-tolerant architecture and redundancy are analyzed. A methodology for assessing the reliability of software (SW) of an automatic control system for technological processes and production is proposed. A mathematical model for assessing the reliability of software is obtained by extending methods for assessing the reliability of hardware to software, taking into account the specifics of the latter. In this case, reliability is understood as the faultlessness of the program, and the main focus is on testing and debugging software in order to identify and correct errors. The mathematical model of software reliability is based on the well-known predictive model, which is based on the assumption of the independence of failures and the exponential distribution of time between failures. The software reliability model takes into account the parameters of the software structure (the number of types, groups of software modules N), the means of implementing the modules (programming language, development environment), which determine the value of the module failure rate; time of formation of the result and the number of detected errors. It is shown that the assessment of the software reliability of the control system is carried out at the design stage when choosing the structure and means of software implementation using expert assessments as model parameters. The mathematical model of the research object allows to assess the reliability of the software: at the design stage, the control system; when choosing the structure and means of software implementation; in the development and debugging of software, as well as in the process of operating the control system.*

Keywords: *software reliability; control system of technological processes and production; elements and devices of computer technology and control systems.*

INTRODUCTION. The technological processes and production reliability is one of the main indicators of the equipment control systems efficiency. It is obvious that the control system failure leads to equipment downtime, technological process disruption and a reduction in the products volume. Control systems for complex equipment complexes in mechanical engineering must have a guaranteed reliability level. In accordance with the standards, reliability is defined as a property that enables a device to perform specified functions with specified characteristics under certain operating conditions and for a required time interval [1].

Modern microprocessor control systems include two main components: hardware; software;
Hardware and software for control systems of flexible manufacturing systems (FMS) and robotic complexes (RC) are complex and unique products. Such control systems reliability is determined using calculation methods, using mathematical models. When assessing the complex microprocessor control systems reliability, as a rule, the hardware reliability is calculated from data on the components reliability.
There are two main methods used to improve the hardware reliability: improving the components reliability; and the use of a fault-tolerant architecture, and redundancy [2-4].
A distinctive microprocessor systems feature is flexible programmable control logic. When assessing the microprocessor control systems reliability, it is necessary to take into account the software failures probability.
In this work, the mathematical model for assessing the software reliability is reduced. It is proposed to extend the methods of the hardware reliability assessing to software, taking into account its specifics. The software failure definition is given in. In accordance with the work, the program reliability is understood as its faultlessness. The main focus is on testing and debugging software in order to identify and fix errors [2-6].
A program is called reliable if its operation algorithm is correct and the program meets the task specifications.
The control system functions scope, implemented with the help of software, tends to increase. The program complication makes it impossible to test it with 100% depth for all possible input and output actions combinations. After complex debugging and operational testing, there may be "hidden" errors in the software.

THE WORK PURPOSE. Improving the microprocessor-based equipment control systems efficiency based on methods for assessing the software reliability.
This goal is achieved by solving the following tasks:
- development of a methodology for assessing the reliability of the control system software based on the data on the reliability of its components;
- estimating the failure probability for software components;
- assessing the control system software reliability based on the proposed methodology.

The following basic statistical and analytical models of software reliability are known [4-7]:
The analytical software reliability model described in the works of Z. Dzhelinski, R. Morand and M. Schuman [4]:

$$R(t) = \exp(-\int_0^t z(x)dx) \quad (1)$$

where R(t) is a reliability function, the probability that no error will appear on the interval [0;t]; z is the density of the error probability distribution. According to this model, the mean time between failures T is equal to:

$$T = \int_0^\infty R(t)dt \quad (2)$$

From formulas (1)-(2) it follows that the average time T between successive error moments detection should increase.

H. Mills [4] proposed a statistical method for assessing the errors number in a program based on testing results. At the beginning of testing, deliberate distortions are randomly introduced into the program codes, which simulate the errors presence. In the program testing process, some of the introduced and own errors are revealed.

H. Mills' software reliability model makes it possible to estimate the number of errors in the program:

$$N = \frac{s \cdot n}{v} \quad (3)$$

where N - the initial number of errors in the program, s - number of introduced errors, v - number of errors entered found, n - number of own errors found.

Another statistical evaluation method [4] tests the program by two groups of developers using independent test suites. The total number of errors in the program is determined by the number of errors detected for each group and the errors that were identified by both groups:

$$N = \frac{N_{12}}{E_1 \times E_2}, \quad E_1 = {}^{N_1}\!/_{N}; \quad E_2 = {}^{N_2}\!/_{N}; \quad (4)$$

where N is a total errors number in the program, N_1, N_2 is a number of errors, discovered, respectively, by the first and second groups developers, N_1 is the number of errors found by both development groups.

Lack of statistical methods for estimating the errors number the program has a low reliability of the obtained results and the need for a large number of tests. Statistical models of software reliability do not predict reliability prior to debugging. Analytical methods for assessing the software reliability use a number of parameters that can only be determined empirically.

Thus, the known reliability models allow getting only an approximate estimate of errors number in the software. These data are insufficient to assess the microprocessor-based equipment control systems reliability.

In this work it is used a mathematical model of software reliability, which uses data about the software structure. This model provides a reliability preliminary estimate already at the design othe control system stage in order to make timely changes to the software structure [5].

Mathematical model of software reliability built on the basis of a well-known predictive model, which is based on the failures and the exponential distribution independence assumption of time between failures [6-9].

Software is viewed as a complex system that consists of individual components, modules. A software component, a module can be a command, a compound operator of a programming language, a procedure, a function, an object, a class, etc.

For each program module, we introduce the following characteristics:

λ_i- is a module failure probability (determined through the failure rate);

P_i- is a failure manifestation probability during module execution.

The failure rate of the module λ_i depends on the way it is implemented (on the task complexity, programming language, development environment, etc.). The failure probability occurring when P_i is executed depends on the software structure [10-12].

We divide the entire set of modules into N groups, types. The N types (groups) number is determined by means of software modules implementation (such as programming language, development environment, etc.) or the software developers’ qualifications, their work experience. Then the software failure rate can be determined through the failure of N type modules rates:

$$\Lambda(t) = \sum_{i-1}^{N} N_i \cdot \lambda_i(t) \cdot P_i(t), \qquad (5)$$

where $\Lambda(t)$ is a software failure rate; $\lambda_i(t)$ is the failure rate of module i-th type; $P_i(t)$ is an error probability occurring when executing module *i*-th type; t is a debug time, N is a number of types, n_i is the number of modules of the *i*-th type.
The P_i coefficient takes into account an error probability occurring when executing a module of *i*-th type. This probability is defined as the error occurrence probabilities product and introducing new errors probabilities during software development:

$$P_i = \left({T_i}/{T_p}\right) \cdot \left({N_i}/{N_p}\right) \qquad (6)$$

where the error probability ${T_i}/{T_p}$ is the ratio of the execution time of the program module T_i to the duration of the operating cycle T_p (or by the time the result is formed); new errors introducing probability when updating the software ${N_i}/{N_p}$ - is the corrections number ratio in the text of a given software module N_i to the total corrections N_p number made during the software revision.
Thus, the following parameters are taken into account in the software reliability model (5):
- software structure (types number, software modules N groups);
- means of implementing modules (programming language, development environment, etc.), which determine the failure rate value of modules $\lambda_i(t)$;
- result formation time T_p;
- number of detected errors N_p.

I. THE RESULT IS THE CONSTRUCTION OF A RELIABILITY MODEL FOR THE SOFTWARE OF TECHNOLOGICAL PROCESS CONTROL SYSTEMS

At the design stage of the control system in the software reliability model (5), analytical (or expert) these parameters estimates can be used. Then, when developing and debugging the software, the parameter values should be refined. During the system operation, the reliability assessment (5) can be corrected taking into account the number of detected errors N_p. As a result of debugging, due to the elimination of errors, the software reliability increases, the failure rate $\Lambda(t)$ decreases.

II. THE DISCUSSION OF THE RESULTS

The computer control system software was tested. During the research object operation, an intermittent failure was noted, with automatic recovery. The refusal manifested itself with some frequency, which was about 7 minutes. If the software failed, it generated an incorrect result once. During the tests, the time between software failures $\Delta t_{отк}$ was determined. Then, in order to fix the error, we made changes to the program. Failure data in the original software code and after two fixes are shown in Tables 1-3.

Table 1.

Mean time between failures after the initial fix in S
(experiment №1)

№ п./п.	1	2	3	4	5	6	7
$\Delta t_{отк}$, мин	7	10	17	24	7	7	21

Table 2.

Mean time between failures after the first fix in S (experiment №2)

№ п./п.	1	2	3	4	5	6
Δt $_{отк,}$ min	14	4	5	9	4	4

Table 3.

Mean time between failures after the second fix in S (experiment №3)

№ п./п.	1	2	3	4
Δt, min	7	10	14	21
t $_{отк,}$ min	7	17	31	52
F(t)	1/4	2/4	3/4	1
p(t)	1/4/7=0,035	1/4/10=0,025	1/4/14=0,017	1/4/21=0,011

After the third fix, the software was tested in different operating modes. The average test time in one mode was 20 min. The total testing time for the revised software version was 6 hours. Failures were not found. The software was found to be serviceable.
Based on experimental data (Table 3) the graphs of the distribution function F (t) and the distribution density p (t) of the operating time between failures (Fig. 2 and Fig. 3), where t is time, min. It can be seen from the graph (Fig. 3) that the distribution density has monotonically form decreasing step function, which can be approximated by an exponential distribution law.

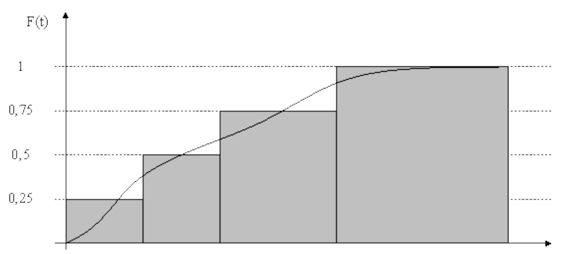

Fig.2. Distribution function graph

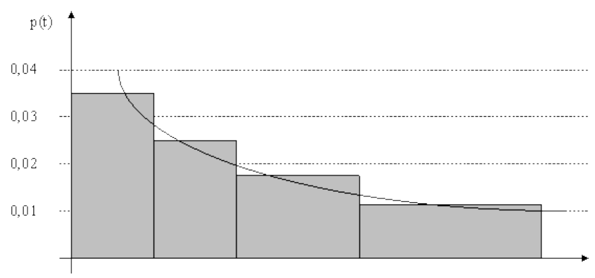

Fig.3. Distribution density plot

Experimental data (Table 1.3) confirm the mathematical software reliability model.

CONCLUSION. A mathematical model is proposed for assessing the equipment control system software reliability. The reliability assessment takes into account the software structure, the means of implementation (programming language, development environment, etc.), formation time of the result, the number of detected errors.
The control system software reliability assessment is carried out:
- at the design stage, when choosing the structu
re and means of software implementation (using expert estimates as model parameters),
- when developing and debugging software using data on software modules and components failures,
- during testing and operation of the control system software
- to assess reliability based on data on failures and the number of corrections in the program text.

References:

1. Yusupbekov, N. R.; Yunusov, B. I.; Gulyamov, Sh. M.; Yunusov, I. I.; and Kasimov, F. A.(2020) "Development Of The Optimal Design Of A Cyclone Apparatus With A Fluidized Bed For The Process Of Metals Pneumatic Separation In The Composition Of Industrial Waste," Technical science and innovation: Vol. 2020 : Iss. 3 , Article 13. DOI:https://doi.org/10.51346/tstu-01.20.3-77-0077 Available at: https://uzjournals.edu.uz/btstu/vol2020/iss3/13.
2. Avrutov V. V. Reliability and diagnostics of devices and systems: Textbook / V. NTUU "KPI", Kiev, 2014. - 156 p.
3. Babakov M.F. Analysis and assurance of the reliability of electronic equipment in the design, Textbook. - Kharkov: National Aerospace University "Kharkov Aviation Institute", 2002. - 100 p.
4. Babeshko E.V. Multistage analysis of reliability and safety of information and control systems / E.V. Babeshko, O. A. Ilyashenko, V.S. Kharchenko // Electronic and computer systems. - 2010. - No. 7. - P. 283–287.
5. Barlow R., Proshan F. Mathematical theory of reliability Per. from English ed. B.V. Gnedenko. - M .: Soviet radio, 1969 .-- 488 p.
6. Yusupbekov A.N., Kh.S.Turaev., Intellectual Decision-Making Support System, International Conference on Intelligent Systems for Industrial Automation, WCIS-2020, Tashkent, Uzbekistan, November 26–28 389-393 p.
7. Lipaev V.V. The quality of the software. M: Finance and Statistics, 1983-C. 227.
   Lipaev V.V. Reliability of software. (Concept review) Automation and telemechanics. - 1986. N10. pp. 5-31.
8. P. Aital and P. Sashikala, "Role of Software Reliability Models in Performance Improvement and Management," Journal of Software Engineering and Applications, Vol. 5 No. 9, 2012, pp. 737-742.
9. Al-Rahamneh Z.A., A New Software Reliability Growth Model: Genetic-Programming-Based Approach, Journal of Software Engineering and Applications, No. 6, 2011, pp. 476-481.
10. Sapaev, X; Abduraimov, E; and Umarov, Sh (2020) "NUMERICAL CALCULATION OF ELECTRONIC CIRCUITS WITH NONLINEAR ELEMENTS," Technical science and innovation: Vol. 2020: Iss. 4 , Article 6. DOI: https://doi.org/10.51346/tstu-01.20.4-77-0095 Available at: https://uzjournals.edu.uz/btstu/vol2020/iss4/6.
11. Uljaev, E.; Narzullaev, Sh.N.; and Erkinov, S.M. (2020) "INCREASING CALIBRATION ACCURACY OF THE HUMIDITY CONTROL MEASURING DEVICE OF BULK MATERIALS," Technical science and innovation: Vol. 2020 : Iss. 3 , Article 23. DOI: https://doi.org/10.51346/tstu-01.20.3-77-0087 Available at: https://uzjournals.edu.uz/btstu/vol2020/iss3/23
12. Karimov, K.A. and Akhmedov, A.Kh. (2018) "DEVELOPMENT AND REALIZATION ON THE ECM OF A MATHEMATICALMODEL OF A FRICTION MECHANISM WITH A CONTROLLED CONNECTION IN THE CONTACT ZONE," Technical science and innovation: Vol. 2018: Iss. 1 , Article 2.DOI: https://doi.org/10.51346/tstu-01.18.2.-77-0004 Available at: https://uzjournals.edu.uz/btstu/vol2018/iss1/2