Impact Factor: 9.2 ISSN-L: 2544-980X

Methods of Testing Logical Control Systems

Abdullaeva Dilnavoz Khusniddinovna [1]

Annotation: Issues of conducting test trials of logical control systems of technological equipment in general and software of systems in particular are considered. Methodology of loading testing of the logical control system core is proposed and number of testing trials, including consumption of CPU resources, consumption of RAM and estimated time of logical control program cycle carried out. Methodologies of calculation of mean time between failures of logical control system are given.

Keywords: logical control, controller, logical control system, testing, mean time between failures.

Introduction

Considering wide range of problems faced by developers of logical control systems (LCS) and, in particular, of machine automation systems, the task of testing and debugging such systems can be noted. Review of various literature sources and scientific works on this topic showed that today there is no comprehensive unified methodology allowing fully and with the required quality to carry out the testing process of control systems and electrical automation systems of machines.

Each newly designed LCS is developed for specific control object, in consideration of features of its operation, user requirements and reliability parameters. Moreover, the system has unified part – the set of modules and components, the presence of which is necessary for interaction with most different types of control objects. Adaptation of logical control systems for a specific control object is carried out by developing a logical control program. This leads developers to the need to make high demands on the quality of debugging of this software. Software testing is not an easy task because this process cannot be fully formalized. Large-scale programs do not have the necessary exact etalon. Therefore, a number of indirect data is used as a guideline, which cannot fully reflect the characteristics and functions of software developments that undergo the debugging process. They are selected so that the correct result is calculated even before the software is tested. Otherwise, some preliminary calculations have approximate character, and if the machine result falls within the expected range, wrong decision about the correctness of the program will be made. [1, 2]

From the point of view of ISO 9126, the quality of software can be defined as the aggregate characteristic of the studied software in consideration of the following components:

- Reliability (impermissibility of failures in operation; the property of the object to maintain operative state for some time or some operating time);
- Tracking (possibility of carrying out the process of improvement, optimization and elimination of software defects after the transfer to operation);
- Practicality (implementation of certain amount of work required for execution, and the individual assessment of such performance by specific or intended circle of users);
- Efficiency (correlation between the level of quality of software functioning and the amount of resources used under specified conditions);
- Mobility (portability from one environment to another);
- Functionality (correspondence of the functionality of the software to the functionality set required by the user).

We will consider a number of practical tasks of conducting testing trials of real logical control systems developed by the team of scientists from "Computer control systems" department of MSTU "STANKIN"| (Russia).

Development of methodology for calculating mean time between failures

Technical parameter called the mean time between failures is used as one of the criteria for the reliability of systems, and characterizes the average duration of the system between repairs and reflects the average time per failure (expressed in hours). Methodology based on the recommendations set in GOST 27.301-95. "Reliability in technology. Reliability calculation. Key points" is carried out in our work. For the core of the logical control system, the mean time between failures will be understood as the time until necessity to completely restart the system. As a criterion, the value of which is estimated and by which a decision is made to reboot the system, we take the cycle time of the logical control core. If the system failures (critical error occurs) or if the cycle time exceeds a permissible value (10 ms for tough real time and 100 ms for soft real time), it is necessary to reboot the system. Reliability indicators are calculated for a specific sample of logical

[1] Doctoral student, Bukhara Engineering-Technological Institute

control systems. The initial data for calculating the reliability indicators are the values of the operating time of the studied sample. [10-12]

To calculate the probability of failure-free operation, the following formula is used

$$P_k = (1-\frac{r_1}{s_1})(1-\frac{r_2}{s_2})(1-\frac{r_k}{s_k}) \quad (1)$$

Where P_k – assessment of the probability of failure-free operation at t_k moment of time r_i – number of failures, S_i – number of objects in the studied equipment sample, operating time of which is more than t_{k-1}. P_k is calculated for each of the intervals t_{i-1} till ti.

The dependence of the probability of failure-free operation on time is determined by the following function:

$$P(t) = \exp(-a_1 t - a_2 t^2) \quad (2)$$

Where a_1 and a_2 – constant coefficients. The function allows predicting the dependence of the probability of failure-free operation on time beyond the time studied during the experiments. Mean time between failures is determined by:

$$T_{cp.} = \int_0^\infty P(t)dt = \int_0^\infty \exp(-a_1 t - a_2 t^2)dt \quad (3)$$

The geometric meaning of Tcp is the area under failure-free operation dependence graph. Next, we consider example of calculating the mean time between failures of a logical control system. For this, tests on five similar prototypes were conducted. For them there is a selection of mean times between failures. The time intervals of the sample are divided as follows: 1000, 1200, 1500, 1700 and 2000 hours. According to the formula, we will calculate the probability of failure-free operation for each of the operating times obtained during the testing trials:

$$P_0 = (1-\frac{0}{5}) = 1 \quad (4)$$

$$P_{1000} = (1-\frac{0}{5})(1-\frac{0}{5}) = 1 \quad (5)$$

$$P_{1200} = (1-\frac{0}{5})(1-\frac{0}{5})(1-\frac{0}{5}) = 1 \quad (6)$$

$$P_{1500} = (1-\frac{0}{5})(1-\frac{0}{5})(1-\frac{1}{5})(1-\frac{0}{5}) = 0{,}8 \quad (7)$$

$$P_{1700} = (1-\frac{0}{5})(1-\frac{0}{5})(1-\frac{1}{5})(1-\frac{0}{5})(1-\frac{1}{5}) = 0{,}64 \quad (8)$$

$$P_{2000} = (1-\frac{0}{5})(1-\frac{0}{5})(1-\frac{1}{5})(1-\frac{0}{5})(1-\frac{1}{5})(1-\frac{1}{5}) = 0{,}512 \quad (9)$$

Figure 1 shows the theoretical graph of the exponential function, constructed according to the formula; the points obtained during testing are highlighted in orange. The results obtained during experimental studies are close to theoretical.

The calculation of the mean time between failures showed that the failure-free operation of the logical control core exceeds 1000 hours, which corresponds to the system requirements. The probability of failure after continuous operation for more than 1500 hours is significantly increased.

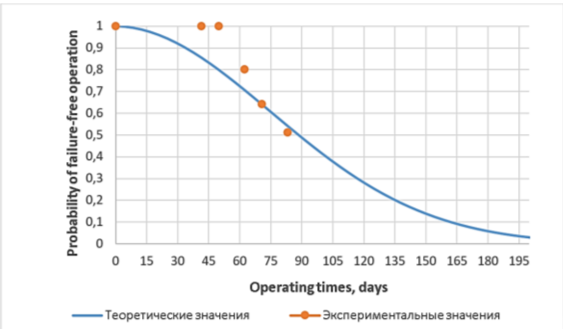

Figure 1. Dependence of the probability of failure-free operation on operating time

The main reasons for the failures were the "hangings" of the logical control core, which is associated with increase in the consumption of computing resources with increase in operating time. As a preventive measure when working with the core of logical control system to eliminate potential failures, it is worth rebooting the system at least once every 1200 hours of operation.

Conclusion

Issues of testing logical control systems were in focus of the work. Load testing was carried out, the purpose of which was to verify the compliance of the system with the requirements. Load testing allowed obtaining an integrated assessment of the quality of the system and evaluating latent defects.

Analysis of results of load testing shows that on all computing platforms exponential increase in consumption of processor resources is observed with increase in complexity of the logical control program. The dependence of the RAM amount occupied on the complexity of the logical control program is linear. In the 64-bit version of the operating system (highlighted in orange on the histogram), the RAM amount occupied by the logical control core is higher than in the 32-bit version by 10-35% depending on the complexity of the logical control program. Cycle time of logical control varies linearly with increasing complexity of the program. The operation of the logical control core under the real-time operating system showed that it is possible to guarantee compliance with tough real-time conditions if the logical control program has less than 20000 functional blocks.

Calculation of mean time between failures showed that the failure-free operation of the logical control core exceeds 1000 hours, which corresponds to the system requirements. The probability of failure after continuous operation for more than 1500 hours is significantly increased. The main causes of failures were the "hangings" of logical control core, which is associated with increase in computing resources consumption.

References

1. Lilija I. Martinova, Sergey S. Sokolov, Petr A. Nikishechkin Tools for Monitoring and Parameter Visualization in Computer Control Systems of Industrial Robots // Advances in Swarm and Computational Intelligence. 6th International Conference, ICSI 2015 held in conjunction with the Second BRICS Congress, CCI 2015, Beijing, June 25-28, 2015, Proceedings, Part II, p.200-207.
2. Georgi M. Martinov, Anton S. Grigoryev, and Petr A. Nikishechkin Real-Time Diagnosis and Forecasting Algorithms of the Tool Wear in the CNC Systems // Advances in Swarm and Computational Intelligence. Volume 9142, 2015, pp 115-126.
3. Georgi M. Martinov, Sergey V. Sokolov, Lilija I. Martinova, Anton S. Grigoryev, Petr A. Nikishechkin Approach to the Diagnosis and Configuration of Servo Drives in Heterogeneous Machine Control Systems // 8th International Conference, ICSI 2017 Fukuoka, Japan, July 27 – August 1, 2017 Proceedings, Part II, pp.586-594.
4. R. A. Nezhmetdinov, S. V. Sokolov, A. I. Obukhov, A. S. Grigor'ev Extending the functional capabilities of NC systems for control over mechano-laser processing // Automation and Remote Control, May 2014, Volume 75, Issue 5, pp 945-952
5. G. M. Martinov, R. A. Nezhmetdinov Modular design of specialized numerical control systems for inclined machining centers // Russian Engineering Research. May 2015, Volume 35, Issue 5, pp 389-393.
6. Martinova, L. I.; Kozak, N. V.; Nezhmetdinov, R. A., The Russian multi-functional CNC system AxiOMA control: Practical aspects of application, Automation and remote control, Vol: 76, No: 1, p.: 179-186, JAN 2015
7. G.M. Martinov, R.A. Nezhmetdinov, and A.U. Kuliev Approach to implementing hardware-independent automatic control systems of lathes and lathe-milling CNC machines // Izv.Vuz. Av. Tekhnika. 2016, no. 2, pp. 128–131. [Russian Aeronautics (Engl. Transl.) vol.59, no. 2, pp. 293-296.]
8. G.M. Martinov, N.V. Kozak, R.A. Nezhmetdinov, A.S. Grigoriev, A.I. Obukhov, L.I. Martinova, Method of decomposition and synthesis of the custom CNC systems // Automation and remote Control (2017) 78: p. 525 - 536.
9. Martinov G.M., Kozak N.V., Nezhmetdinov R.A. Implementation of Control for Peripheral Machine Equipment Based on the External Soft PLC Integrated with CNC // 2017 International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM), 16-19 May, 2017.p.1-4
10. Martinov, G., Kozak, N. and Nezhmetdinov, R. (2018). Approach in Implementing of Logical Task for Numerical Control on Basis of Concept "Industry 4.0". 2018 International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM). ISBN**:** 978-1-5386-4307-5
11. Nezhmetdinov, R., Nikishechkin, P. and Nikich, A. (2018). Approach to the Construction of Logical Control Systems for Technological Equipment for the Implementation of Industry 4.0 Concept. In: 2018 International Russian Automation Conference (RusAutoCon). Sochi: IEEE.
12. Kovalev, I., Nezhmetdinov, R. and Nikishechkin, P. (2019). Approach to Assessing the Possibility of Functioning of CNC and PAC Systems on Various Software and Hardware Platforms. In: 2019 International Multi-Conference on Industrial Engineering and Modern Technologies (FarEastCon). Vladivostok: IEEE, pp.1-5.