ISSN NO: 2770-4491
Date of Publication:10-06-2022

Approach to Testing Logical Control Systems of Technological Equipment

Abdullaeva Dilnavoz Khusniddinovna - Doctoral student
Zainiev Khudoyberdi Mukhiddinovich – Probationer teacher
Bukhara Engineering-Technological Institute

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.

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).

ISSN NO: 2770-4491
Date of Publication:10-06-2022

Development of methodology of load testing the core of logical control system

By load testing we will mean testing the performance of control system, in which collecting the indicators, determining the response time of system and performance in response to external influences will be carried out [3-5]. The purpose of load testing is to verify that the system meets the requirements. Load testing allows getting an integrated assessment of system quality and evaluating hidden defects. Another purpose of load testing is to monitor system performance at given load. Generally, load testing is performed for multi-user systems in order to determine the number of simultaneously serviced customers without loss of quality. In the case of a logical control system, it is advisable to evaluate the dependence of the used computing resources on the complexity of the system. [6, 7] Several indicators, including, can characterize the complexity of the system: the number of hardware-controlled inputs/outputs, the number of functional blocks used in logical control program. Part of the computing resources is consumed to calculate each block in the control cycle, the complexity of the program increases with increase in the number of functional blocks and the consumption of computing resources increases, so it is advisable to use this indicator as the base indicator for carrying out load testing of the logical control syste. Load testing is necessary to formulate performance requirements for the computing core at the stage of development of functional requirements. Often, it is not possible to clearly formulating requirements before the development of a management system. In this case, it is necessary to analyze the complexity of the system being developed and make assumption about the expected load and consumption of hardware resources, based on which to conduct exploratory load testing.

In the case with a logical control system, the core, as the application most critical to the available computing resources, was subjected to load testing. For load testing, the following application performance indicators were selected:

- CPU resources consumption (estimated in percentage). Indicates the relative time spent by the processor on the calculations for the needs of the selected process. Currently, processes can operate in several flows, which allows the processor to perform parallel computations. The obtained metrics of processor resource consumption allows analyzing the effect on the system performance of the number of flows, application configuration, type of operating system, its settings, etc.
- RAM consumption (estimated in MB). The indicator estimates the absolute value of the memory consumed by the application. Monitoring of this indicator allows optimizing the memory used by the application. While the application is running, there are links to objects in memory, and a process called the Garbage Collector (GC), which also uses processor resources, clears unused links. Moreover, if the large amounts of memory is allocated to application, the time spent by the processor can be significant. During the cleaning of the memory pages, the access of the main application processes to them is blocked, which affects the end time of the application. GC is not used at the core of the control system.
- Cycle time of the logic control program (estimated in ms). It is the most critical indicator of the performance of a logical control system, as directly characterizes real-time processes.

For the core of the control system considered in the work, testing was carried out at the stage of early development in order to determine the correctness of selected architectural decisions in the design. Such testing is called proof-of-concept testing.

The statistical data obtained during testing are in the form of tables with large amounts of numerical data, and are difficult to analyze. Moreover, the properties of the data obtained are often difficult to identify, because presentation of information in the form of big data arrays is not clear and does not give a visual characteristic of the presence of patterns. In the work, we used the histogram tools for the analysis of load testing data, which are most often used in statistical analysis methods.

CPU resources consumption. We will consider test scenario for CPU resources consumption testing. Number of logical control programs have been developed for testing, which have a different number of functional blocks used: 100, 850, 1500, 3500, 5000, 7500, 10000, 12500, 15000, 17500 and 20000 blocks. The size of logical control program of 20000 functional blocks, selected as the maximum, is practically not achievable when programming real objects. The test results are shown in Figure 1.

Four computing platforms were used during testing: personal computer Core i7 processor with a frequency of 1.8 GHz (highlighted in orange on the histogram), laptop Pentium Dual processor with a frequency of 2.2

ISSN NO: 2770-4491
Date of Publication:10-06-2022

GHz (highlighted in blue on the histogram), single-board computer Orange Pi (highlighted in gray on the histogram), and single-board computer Odroid XU4 (highlighted in yellow on the histogram). The personal computer meets the requirements for computing nodes of automation of technological equipment. The laptop allows assessing the dependence of the performance of the computing platform on the form factor. Two single-board computers are selected in order to assess the load capacity of mobile platforms of various performance.

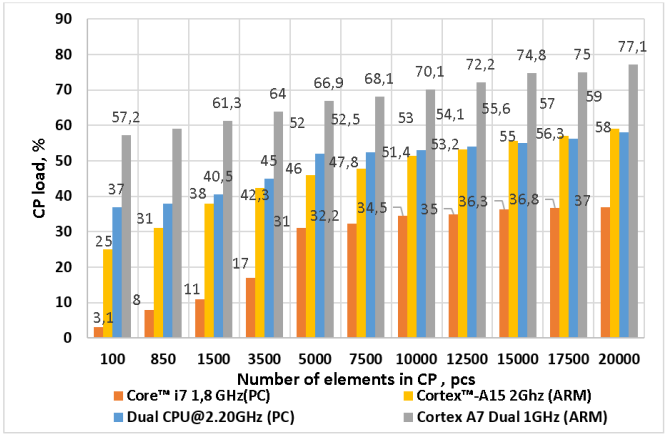

Figure 1. Dependence of CPU resources consumption on program complexity

Analysis of load testing results shows that on all computing platforms, exponential increase in the consumption of processor resources with an increase in the complexity of the logical control program is observed. On the least efficient single-board computer with ARM architecture at the program of 20000 functional blocks, the processor was loaded at 77%, which allows concluding that all four platforms selected for testing can be used to implement logical control systems. However, the decision about which computing platform is more appropriate to apply should be made depending on the complexity of the system. Load testing also showed that the processor clock speed does not directly affect the amount of processor resources consumed. This suggests that the hardware computing devices used in the system should be as balanced as possible in terms of all performance characteristics. Assessing the relative increase of amount of processor resources consumed on each of the computing platforms, we can conclude that the average growth was 25-30%. From this we can conclude that the most significant factor affecting the consumed processor resources is not only the complexity of the logic control program algorithms, but also the amount of resources used by the core application itself to serve internal needs.

RAM consumption. The core of logical control system can operate under the control of both 32-bit and 64-bit operating systems. However, 64 - bit version of the operating system uses a lot more memory due to the fact that the variables with which it operates are not 32-bit, but 64-bit. When designing a system, it is important to evaluate whether using a 64-bit version of the operating system leads to a critical increase of RAM amount used. RAM consumption was estimated for two identical computing platforms (PC with a Core i7 processor with a frequency of 1.8 GHz), but under the control of operating systems with 32-bit and 64-bit (Figure 2).

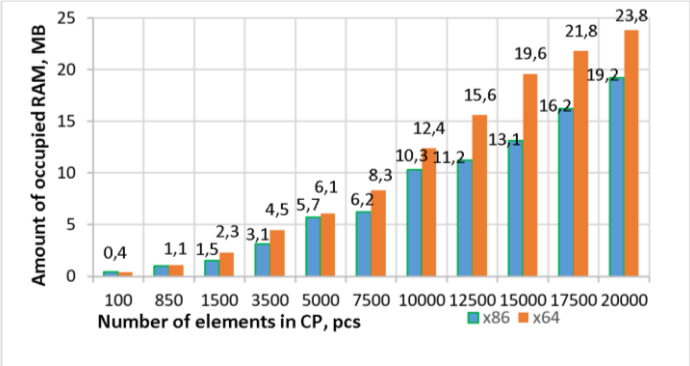

Figure 2. Results of testing RAM consumption

ISSN NO: 2770-4491
Date of Publication:10-06-2022

The results of load testing showed that the dependence of the amount of RAM occupied on the complexity of the logical control program is linear. In the 64-bit version of operating system (highlighted in orange on the histogram), the amount of RAM 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 program. To assess the reliability parameters of the system, it is necessary to determine the degree of influence of the complexity of the logical control program on the time of the control cycle; for this, tests were carried out using logical control programs of varying complexity (from 100 to 20000 elements). The test results are shown in Figure 3. Testing was conducted in aim to assess the impact of the type of operating system on the time of the logical control cycle, so the tests were conducted under operating systems that use machine time (Windows XP - blue on the histogram, Ubuntu 11.04 - orange on the histogram) and real time (Linux RT - gray on the histogram).

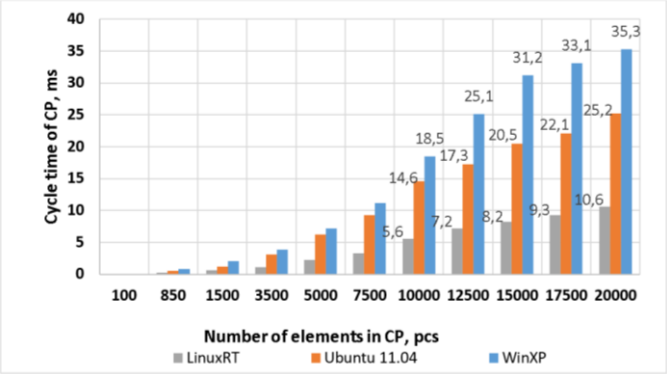

Figure 3. Dependence of control cycle time on the complexity of the program

The 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 for guaranteed operation in tough real-time mode, the logical control program should not exceed 20000 functional blocks. The operating systems of the Windows and Ubuntu families can only be used for debugging logical control programs or for educational purposes, because do not satisfy the operating conditions of the system in real time.

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 control systems. The initial data for calculating the reliability indicators are the values of the operating time of the studied sample.

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.

ISSN NO: 2770-4491
Date of Publication:10-06-2022

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.].