Spacecraft onboard computers control automation

E V Tabakov$^{1,2,4}$, A I Zinina$^2$ and M Yu Kosinskiy$^{1,3}$

$^1$Design Bureau «Mars», 16 1st Shemilovsky lane, building 1, Moscow, 127473, Russia
$^2$National Research University Moscow Aviation Institute, 4 Volokolamskoe avenue, Moscow, 125080, Russia
$^3$National Research University Power Engineering Institute, 14 Krasnokazarmennaya str., Moscow, 111250, Russia

$^4$E-mail: e-tab@rambler.ru

Abstract. This paper concerns the task of onboard spacecraft’s computers functioning analysis problem. Existing approach is analyzed. Criteria of onboard computers normative functioning and source data for analysis implementation are defined. As a result, telemetry analysis algorithm for forming decisions about onboard computer normative functioning is offered. Program based on it is developed.

1. Introduction

The most important part of the spacecraft is its control system (CS) [1]. CS is based on the onboard computer (OBC) [2]. OBC solves tasks of supporting spacecraft functioning and performs its mission. During the CS operation OBC implements execution of the software that allows CS to solve its tasks [3]. This software is divided into the system one and the functional one accordingly to the solving problems [4]. The functional software (FS) is responsible for the spacecraft mission execution. The system software or computational process supporting system (CPSS) in other words is the software complex that provides computational process execution. One of the CPSS tasks is OBC functioning control and diagnostics.

During the ground test process as well as spacecraft operation at the orbit OBC operates in different modes: ground testing mode, flight mode and computer reconfiguration mode.

Comprehensive computational systems simulations are executed during ground CS tests to confirm its accordance to the given requirements [5]. Ground tests are carried out according to special technology. This technology includes both functional and hardware components tests. During the tests of the CS functional components control algorithms logic is investigated mostly. Electric interactions between CS hardware components aren’t analyzed at this stage. Functional components tests are executed at hybrid system testbed (STB). The OBC is connected to the simulator which is simulating the satellite's equipment. Additionally the still missing analog, digital, serial, pulse and bus interfaces of the OBC are simulated. Functions of the onboard software can be tested while running partially on the real hardware and connected with the simulated equipment. At STB OBC operates in flight mode [6].

At the electrical functional model testbed (EMF) complex CS checkout is implemented focusing at hardware interaction with other satellite systems. Mainly, at the EMF real hardware is used. The
hardware operation in the flight mode often isn’t possible. In this case, at EMF, OBC operates in the ground mode.

CPSS has means of control and diagnostics system (CDS) that executes the OBC operation analysis. Particularly these means implement the expanded integrated control test (EICT). EICT is always performed at the moment of OBC turning on and includes checks of processor command, special mission registers, timers, random access memory (RAM), root and page reprogrammable memory.

The PM execution period compliance to the given time limits is the important thing also. It characterizes the correctness of the OBC operation. It should be noted, that the analyzed CS is a «hard» real time system. The excess of the given time limits for program module (PM) operation time is prohibited in such systems, because it may cause the CS malfunction. The operation times are also fixed by CDS. So there’s an important task of the PM execution time compliance to the given time limits check.

Normative PM operation period limits are defined at the OBC cyclogram for every mode of operation.

OBC includes four similar programmable logical controllers (PLC). But only one PLC may be the master device for other spacecraft systems. The first in order or the most workable PLC is chosen as the master one. Herewith two other PLCs operate as a «hot» reserve (to inspect the correctness of master PLC computations) and the one PLC becomes the «cold» reserve (turn off). CDS forms the value of parameter that denotes OBC’s configuration.

Values of parameters that characterize the OBC operation are recorded in telemetric information (TMI) by CDS.

Thus there is an opportunity to perform OBC normativeness analysis by comparison the values of parameters that characterize OBC functioning to their nominal values that are defined by the mode of its operation. However, TMI consists of enormous quantity of parameters that characterize FS and CPSS operation. So it complicates the analysis execution.

Previously OBC operation analysis was implemented manually by comparison of each parameter value with its nominal.

However this approach takes a lot of time, material and labor resources, so it is extremely inefficient. In addition, there is a significant probability occurrence of errors due to human factor while using this approach.

So there is a task of such analysis automation. We decided to perform analysis of OBC turning on parameters values at first step. It will make possible to get a solution in a rather short time. The turning on phase is extremely important for OBC operation. At this moment the parameters that indicate further OBC operation normativeness are formed.

2. Source data
The values of the parameters that characterize the correct OBC operation in various modes, and the results of its extended built-in control test (BCT) are the source data for the developed OBC normativity assessment algorithm.

Onboard computer operation cyclograms are described in documents of the standard format. A special internal standard file that contains information about the OBC cyclograms is created based on these documents. This file contains the information for the analysis process.

Information about the PM operation times is recorded in the TMI as arrays of actual PM completion time in the current computation cycle for every PLC of the OBC.

The number of operating PLC is determined by the value of a special parameter, where each digit determines the on / off state of the corresponding OBC PLC.

All the information that is contained in the TMI has a format that includes:

- Parameter generation time;
- Parameter identifier;
- Parameter value.
3. Proposed algorithm
The computer operation mode is determined by the value of the parameter that characterizes the link with the ground equipment. The nominal parameter values that characterize the onboard computer functioning are set based on the mode of the OBC operation.

The analysis of the OBC normative functioning is executed by comparison of the actual parameters values in the TMI with their nominal ones for the current OBC mode. If the parameter value differs from the corresponding nominal one, a conclusion about its non-normativeness is formed and the record about it is made to the database.

The OBC PM execution times normativeness analysis consists of verifying the conformity of actual execution times to the required ones. At present, the PM normative execution criterion is that the actual time of its operation does not exceed the specified part of the time interval accordingly to the OBC cyclogram. The results of this analysis are the PM maximum operation time for every OBC PLC, the actual PM execution time, the PM completion time and the PM execution time accordingly the cyclogram. Further, time reserve is computed both as absolute and relative values.

The volume of the analysis is determined by the amount of available information.

The general conclusion about the OBC normative operation is formed based on the results of the particular analysis: deviation from the normative value even for one parameter or execution time exceeding of the given time limit even for even one PM one PM are unacceptable. In this case a conclusion about the OBC failure is formed.

Special software tool was developed to solve practical tasks based on this algorithm.

4. Software implementation
The application named «Cyclogram» is designed based on the algorithm described above.

The programming language is chosen C++ in order to provide consistency with the existing software at the enterprise. The framework of development is chosen QtCreator. This framework has been chosen due to its widespread distribution at the enterprise in the processes of graphical user interface applications development.

It should be noted, that the enterprise has already automated the process of tests implementation. The Test Automation System (TAS) is one of its the most important aspects. This system allows to automate the loading of the source data, starting of analysis and recording of results to the database. So we integrated our application in the TAS.

So the «Cyclogram» application uses the TMI that is loaded from TAS database to process the analysis. As a result, we get a report with normativeness conclusion description.

The report has an .ods file type and may be opened with the OpenOffice Calc program as a table. The .ods file format has been selected to present the results because it is widely spread. Also there are several free office applications (OpenOffice, LibreOffice) for comfortable work with it and simple methods for its automatic handling.

The special method was proposed for automating the creation of * ods format report files. ods files may be created from xml files using the archiver.

So normativeness analysis results are saved as xml files first and are converted into ods files after. The open source TinyXml library was used for this. Its capabilities are fully suitable for solution of this task.

The developed application has two operating modes: graphical user interface (GUI) mode and hidden-mode (via the command line).

5. Software work description
The main window graphical user interface is shown in figure 1. This interface provides tools for setting the path to the necessary files and starting the analysis.
Figure 1. The graphical user interface of developed application.

Figure 2. The informational message window when cyclogram file wasn’t analyzed.

An informational message is the result of analysis for graphical mode (figure 2). It contains the main results of the analysis.

The extended protocol file is created if there is data for PM execution times normativeness analysis. It is shown in figure (figure 3).

Rows of the table are automatically colored in two alternating colors in order to improve perception of information in the report.

The figure 4 shows the protocol view for the case when the several PMs execution time exceeds the given time limits. Rows of the table, corresponding to these modules, are highlighted in red.
6. Conclusion
As a result of the work, an algorithm for automating the OBC operation analysis has been proposed. It allows to control OBC operation. The actual (e.g. registered during the test execution) values of the parameters that characterize the correct OBC operation in various modes, and the results of its extended BCT are compared to correct ones. Also it allows to analyze the normativeness of actual PM operation times.

The software was created based on the proposed algorithm. The application was integrated into the existing enterprise test automation system. So, the automation of the OBC operation analysis was greatly simplified.

References
[1] Shen Q, Yue C, Goh C H and Wang D 2019 Active fault-tolerant control system design for spacecraft attitude maneuvers with actuator saturation and faults IEEE Transactions on Industrial Electronics 66(5) 3763-72
Sternberg D, Essmiller J, Colley C, Klesh A and Krajewski J 2019 Attitude control system for the Mars cube one spacecraft IEEE Aerospace Conf. (Big Sky, MT, USA) pp 1-10
Peng T et al. 2016 A Component-Based middleware for a reliable distributed and reconfigurable spacecraft onboard computer IEEE 35th Symposium on Reliable Distributed Systems (SRDS) (Budapest) pp 337-42
Eickhoff J 2011 Onboard computers, onboard software and satellite operations: an introduction Springer Science & Business Media p 16
[3] Leveson N 2004 Role of software in spacecraft accidents Journal of spacecraft and ROBCkets 41(4) 564-75
[4] Sinicyn S V and Grigor’eva O L 2019 System Onboard Software. Operational Development Environmen vol 3 (Moscow: MOKB «MARS») p 43
[5] Nozhenkova L, Isaeva O and Vorogovskiy R 2017 Scenario approach to testing spacecraft’s onboard equipment command and software management DESTech Transactions on Engineering and technology research (Beijing: DESTech) pp 53-7
[6] Andeeev V P, Volovich N V, Glebov I V et al. 2011 Designing and Testing Onboard Control Systems (Moscow: MAI-PRINT) pp 151-6