Modelling of the Spacecraft onboard apparatus and ways of building a consistent control logic in case of limited onboard resources

A.A. Tyugashev[1], Yu.M. Sygurov[2]

[1]Samara State Transport University, Svobody 2V, Samara, Russia, 443066
[2]Space Rocket Centre 'Progress', Zemetsa 18, Samara, Russia, 443009

Abstract. The paper describes the model of the control algorithm for such Physical (Technical) complex system as modern spacecraft. The mathematical models for onboard systems and devices as well for onboard flight control software is being proposed. Each onboard device, aggregate, sensor consumes some resources. The paper discuss problem statement of building consistent real-time control logic that guarantees completion of spacecraft mission in case of limited onboard resources. The method and software tool is being described.

1. Introduction

The onboard control complexes of existing and prospective Earth observation spacecrafts is based on wide use of computer systems and onboard computer networks. These computers running under control of large onboard flight control software [1,2,3]. The main part of onboard control complex is an onboard computer system and the organizing core of onboard flight control logic is algorithms implemented in flight control programs [3]. During both pre-mission testing and space operations we face with not equipment failures and damages only but also with improper functioning of flight control software or software embedded in some onboard device. As a rule, such abnormal situations are not expected at spacecraft's design stage and analyzed in operational documentation. Consequently, it is impossible to parry these situations during the flight.

On the other hand, the growing requirements to tasks to be executed by automatic spacecraft, lead to increase of number of onboard devices and level of complexity of control logic. Correspondingly, the complexity of flight onboard software increases dramatically [1,4].

Another important factor we need to consider is restricted onboard resources. Each onboard device consumes some onboard resource during functioning, for example, electric power. We also can mention memory, shared central processor time slots, etc.

Each onboard device has a number of working modes with different levels of consuming of different kinds of resources. At least, each onboard device has two modes: ON and OFF. Specific devices can have a wide spectrum of modes with varying consumption of resources. Depending on the current flight conditions, some equipment can be switched on or shut down over time. But there are some 'red lines' (maximal available value, limit) of levels of each kind of onboard resources which the control algorithm should not violate during whole flight despite the current onboard situations, both normal and abnormal. For example, there are solar panels which provide spacecraft with certain limited level of electric power.

Device transition from mode to mode is being implemented by the so-named “apparatus control commands” (coded sequences of electric impulses transferred by onboard cables). These commands issued by real-time control algorithms implementing spacecraft control logic. For example, we can deal with the command «SWITCH ON Device1 in System1» or «CHANGE MODE OF GyroDyne1 to Mode2».Today we need the means allowing design of robust control logic providing confidence in non-violating of available resource levels but at the same time guarantees fulfillment of spacecraft mission.

The onboard control logic of modern automatic spacecrafts is being implemented by set of control algorithms. In turn, the particular control algorithm is being implemented by concrete module of onboard software. The program module contains various operators, but from control logic point of view we interested in only the following actions: 1) call of other module implementing other control algorithm (the very important issue is that program modules is running under control of multi-thread onboard operating system, so we deal with parallel execution of several control algorithms) 2) issue of control command to particular device; 3) Setting or reset of some predicate (flag, logical condition). Then values of these predicates influence on execution of other control algorithms. For example, the condition can has the following sense: «second solar panel is fully operational» or «level of charge of onboard battery is less than 10%». Sometimes these predicates are named as ‘flags’ which can be set or reset. The set of all these conditions forms the ‘spacecraft state vector’. If we fix the certain values of the predicates, we get the concrete scenario of execution of the control algorithm.

Herewith, different scenarios of control algorithm’s execution lead to execution of different sequences of actions with certain time stamps depending on values of some logical conditions (predicates).

Each action in control algorithm is associated with certain moment of time. In other words, we can speak about ‘cyclogram of actions’ to be executed or even set of cyclograms for different scenarios.

The complexity of design in case of limited resources problem is due to the following circumstance. As it was mentioned above, the several control algorithms can be running in parallel executing different sequences of actions. And we deal with the chain (in fact, tree) of calls CA1->CA2->CA3… . One control algorithm can call (activate) dozens of other control algorithms. It is a very hard problem for a designer to consider all these actions especially with the control of non violation of resource limits.

This paper is devoted to some approaches to solving of named problem.

2. Problem Statement

As it was mentioned above, we have:

$EA = \{BA_i\}, i=1...N$ – set of onboard devices;

$AP = \{AP_j,\} j=1..M$ – set of onboard resources;

Lim $(AP_j) = \{ (LimAP_1, LimAP_2,... LimAP_M) \}$ – set of maximum available levels (limits) for each kind of resource;

REA $(BA_i) = \{ (RM_{i1}, RM_{i2}... RM_{iN}) \}$ – vector of modes of functioning corresponding to each onboard device;

$CL (RM_{ik}) = \{ (CP_{i1}, CP_{i2}... CP_{iN}) \}$ – vector of levels of consumption of each kind of resource in certain device’s mode;

The starting point in control logic design process is set of tasks to be executed by the spacecraft. The important issue is that we have required time moment, corresponding to moving in space and other physical processes onboard for each task.

$Z = \{ PT_j,t_j \}$ - set of tasks to be executed by spacecraft’s equipment.

$UA = \{ CA_m \}, m=1...U$ – set of control algorithms.

We need:

1. Determine mapping

$$f_1: Z \rightarrow EA \quad (1)$$

i.e. define which device is used in processes of certain task’s execution, and which modes of functioning of this device we need.

2. Determine mapping:

$$f_2: EA \rightarrow UA \quad (2)$$

i.e. mapping between elements of onboard apparatus and algorithms which control them.
3. Set time parameters:

$$f_3: EA \rightarrow T \text{ and } UA \rightarrow T \quad (3)$$

Based on this, we can build the cyclogram of functioning of various onboard devices as well as calculate the level of consumption of all kinds of resources as function of time. To have an imagination about the cyclogram, .

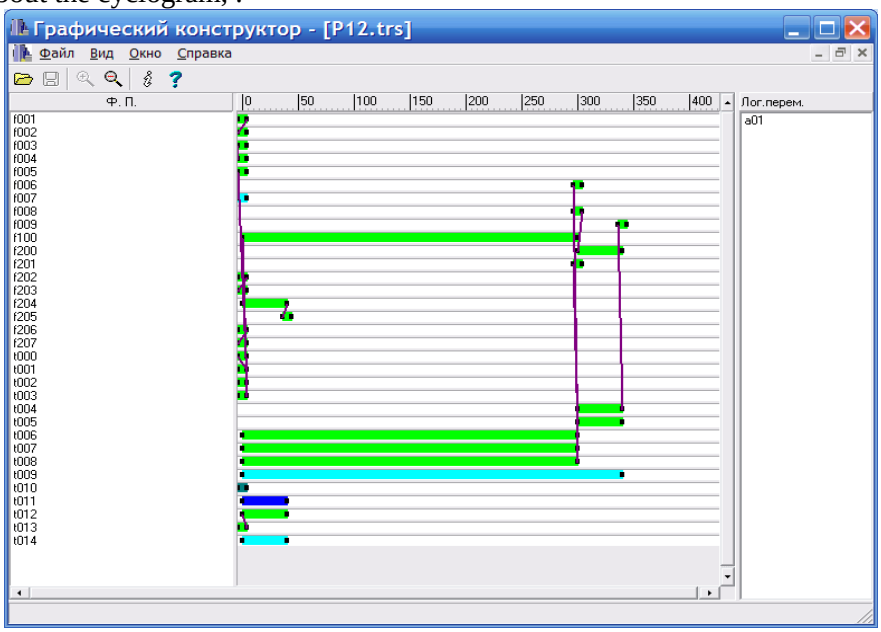

Figure 1. Cyclogram of onboard operations (screenshot of soft tool).

The last mapping allows specifying the following tuples:

$$\{BA_i, CA_{ij}, t_i, l_i\} \quad (4)$$

Each tuple defines for the each onboard device: the control algorithm CA_i and scenario of its execution (branch) j starting at time moment t_i in situation reflected by logical condition vector l_i.
The logical vector can be represented, for example, as $l=(\alpha_1=TRUE, \alpha_2=FALSE, \ldots \alpha_Q=H)$ where α_l is a particular condition. Some conditions in particular moment of time can be undefined or irrelevant, in this case we use 'H' value of truth (one can say, in this aspect our approach is similar to some kind of three values logic).
The set of these tuples allows building the ordered sequence W of 'sections' of spacecraft flight and to draw the diagrams showing the different scenarios of implementation of control logic. Also the W sequence can be used for checking of overlaps of device modes and program modules work. Moreover, linking the work of control algorithms with functioning of onboard apparatus (systems and particular devices) we can get the time diagram (cyclogram) of onboard equipment's work and the consumption schedule for certain onboard resource.
Alternate approach which can be applied [1,4,7] is based on following tuples:

$$CA_m=\{ <f_i, t_i, \tau_i, l_i>\}, f_i = Call(CA_k) \vee KU(BA_i) \vee Set(\alpha_l), \quad (5)$$

where f_i is an action to be executed at time moment t_i with duration τ_i and associated with logical vector, l_i. f_i might be call of other control algorithms, issue of particular control command to onboard device or setting of some component of spacecraft's state vector.

2.1. Stages of control logic design

To provide some means (with some level of automation) for supporting of spacecraft control logic design we need to step-by-step analyze stages of design and the information used at each stage.
The process of design of control logic supported and implemented by onboard apparatus and flight control software must allow harmonizing functioning of different onboard systems and units in physical and logical senses. In physical sense, for example, we need consider the maximum available

levels of onboard resources. At this stage, the base cyclogrammes of onboard apparatus and units work should be built. So, we need to build the certain logical procedures both for control of particular onboard device (control 'in small') and for control of spacecraft as a whole (control 'in big'). These procedures should be implemented by control algorithms for complex operation of spacecraft [4,5] at the next stage. At this stage, the following information must be considered:

- materials on the control logic of systems and units during execution of functional tasks;
- requirements to time of functional tasks' execution including requirements on possibility or restriction of simultaneous execution (mutual overlay) of various functional tasks;
- requirements to the sequence of execution of different sections of control algorithm.

As a result, we will get the following stuff:

- input data for development of control algorithms for complex operation;
- time diagrams showing the operation of systems and units, indicating the modes of operation of the equipment and algorithms for different scenarios of spacecraft mission realization;
- materials on mutual overlay of algorithms and modes of onboard device functioning;
- estimation of required energy and other kind of onboard resources consumed by spacecraft equipment during operations.

2.2. Formal mathematical models for input and output data of different design stages

To build the formal representation of models for input and output data of different stages of design of spacecraft's control logic we need first consider the structure of the model which will allow to describe the named entities related to different kinds of onboard systems, devices and aggregates with the specifying of work modes. On the other hand, the model should describe the set of control algorithms divided into the 'sections' and with the different scenarios of execution (branches). The 'core' part of onboard flight control software is so-names 'control algorithms for complex operations'. The model of such kind of algorithms can be reviewed as a 'base' model.

We propose the use of the following additional sets:

$EA = \{BA_i\}$ – set of onboard devices;

$A = \{ CA_{ij} \}$ – set of control algorithms where the i is ID of the algorithm, and j is number of possible scenario (branch or path of execution). As a rule, each control algorithm has several (Ki) scenarios of execution depending on truth of certain logical conditions. So, CA_{ij} is a control algorithm with ID i implementing on scenario j.

$\Omega = \{ \omega_l , < \}$ – ordered (in time) set of sections of spacecraft operation.

We can describe the particular control algorithm by the following structure:

$CA_{ij} = \{ T^{ij}_{work}, A^{ij}_{on}. A^{ij}_{off}, BA_{ij}, KU_{ij}, FP_{ij} \}, i=1..N, j=1..K.$

Here $T^{ij}_{wor} = \{(tw^{ij}_1,lw^{ij}_1), (tw^{ij}_2,lw^{ij}_2), \ldots (tw^{ij}_{Ki},lw^{ij}_{Ki}) \}.tw_{ij}$ is a start time of execution of control algorithm scenario in case of truth of logical condition lw^{ij}. The number of pairs $(tw^{ij}_{Ki},lw^{ij}_{Ki})$ defines the number of calls of scenario j of control algorithms depending on lw^i. The full set of logical conditions lw^{ij} defines the 'state vector' of the spacecraft [6,7].

A^{ij}_{on} is a set of other algorithms called by scenario j of CA_{ij}. $A^{ij}_{on} = \{(CA^{ij}_1,tw^{ij}_1,lw^{ij}_1), (CA^{ij}_2.tw^{ij}_2,lw^{ij}_2), \ldots (CA^{ij}_K,tw^{ij}_K,lw^{ij}_K) \}$. Tuple means $(CA^{ij}_K,tw^{ij}_K,lw^{ij}_K)$ that our control algorithm calls algorithm CA^{ij}_K at time tw^{ij} if lw^{ij}_K is true.

Similarly, A^{ij}_{off} is a set of other algorithms which are being shut down by scenario j of CA_{ij}. $A^{ij}_{on} = \{(CA^{ij}_1,tw^{ij}_1,lw^{ij}_1), (CA^{ij}_2.tw^{ij}_2,lw^{ij}_2), \ldots (CA^{ijj}_K,tw^{ij}_K,lw^{ij}_K) \}$. Tuple means $(CA^{ij}_K,tw^{ij}_K,lw^{ij}_K)$ that our control algorithm shuts down algorithm CA^{ij}_K at time tw^{ij} if lw^{ij}_K is true.

BA_{ij} is onboard equipment controlled by this algorithm. $BA_{ij} = \{(Nam_{ij},R_{ij},P_{ij})\}$ where Nam_{ij} is a name (ID) of particular onboard unit or device, R_{ij1} is a mode of work of this device, and P_{ij} is a vector of consumption of available onboard resources in mode R_{ij}.

KU_{ij} is a set of commands (device control code sequences, impulses) issued by control algorithm to the particular onboard device or unit. $KU_{ij} = \{(NK^{ij}_1,tw^{ij}_1,lw^{ij}_1), (NK^{ij}_2.tw^{ij}_2,lw^{ij}_2), \ldots (NK^{ij}_K,tw^{ij}_K,lw^{ij}_K)\}$ where NK^{ij} is a name (ID) of control command, tw^{ij} is a time of command issue in case of truth of lw^{ij}_K.

$FP^{ij} = \{(PI^{ij}_1, ts^{ij}_1, ls^{ij}_1), (PI^{ij}_2, ts^{ij}_2, ls^{ij}_2), \ldots (PI^{ij}_k, ts^{ij}_k, ls^{ij}_k)\}$ is a set of flags (predicates) formed by j scenario of algorithm CA_{ij} execution at time moments ts^{ij}_k in case of truth of condition ls^{ij}_k.
The presented models contain information required to determine the sequence of sections of spacecraft control logic and to form diagrams showing spacecraft functioning in different scenarios and from various points of view. Also we can determine overlays between algorithms and functioning of certain onboard equipment.

3. Conclusion and Future Work

The executed analysis of the process of spacecraft control logic design has shown the existence of opportunity of formal representation of data related to different stages of design, in form of mathematical models (and electronic models used in special instrumental design software tools). Now the authors work on implementation of presented models in software tools by electronic means.
The goal of information system for support of design of spacecraft control logic in case of restricted onboard resources can be formulated as providing designer with the tools allowing automatically calculate estimated levels of consumption of onboard resources based on formal models of control algorithms. These estimations should be found for all time period of spacecraft mission. The other important required feature is forming and visualization of cyclograms showing execution of tasks, calls of program modules, functioning of certain equipment.
Supposed introduction of these tools into spacecraft control logic design process at Space Rocket Centre 'Progress' will help to support decision making, allow to optimize the cyclograms of functioning of onboard apparatus, reduce labor costs, increase the quality of design documentation. These aspects are quite actual and prospective in modern conditions when we face with the growing importance of information models and complexity of software. We can claim that this future work has the significant practical impact.

4. References

[1] Tyugashev, A.A. Integrated environment for designing real-time control algorithms // Journal of Computer and Systems Sciences International. – 2006. – Vol. 45(2). – P. 287-300.
[2] Kozlov, D. Control of Earth observation spacecrafts: Computer Technologies // D.I. Kozlov, G.P. Anshakov, Ya.A. Mostovoy, A.V. Sollogub. – Moscow: Mashinostroenie, 1998. – 245 p.
[3] Tyugashev, A.A. Language and Toolset for Visual Construction of Programs for Intelligent Autonomous Spacecraft Control // IFAC, 2016 (Papers OnLine).
[4] Kalentyev, A.A. CALS technology in lifecycle of complex control programs / A.A. Kalentyev, A.A. Tyugashev. – Samara: Scientific Center of Russian Academy of Sciences, 2006. – 266 p.
[5] Filatov, A.V. Structure and algorithms of motion control system's software of the small spacecraft / A.V. Filatov, I.S. Tkachenko, A.A. Tyugashev, E.V. Sopchenko // Proceedings of International Conference Information Technology and Nanotechnology ITNT, 2015. – P. 246-251.
[6] Kalentyev, A.A. Development of information support for the spacecraft controlalgorithm design process / A.A. Kalentyev, Yu.M. Sygurov // Vestnik SGAU. – 2010. – Vol. 21(1). – P. 58-62.
[7] Bogatov, A.Yu. The logical calculus of control algorithms / A.Yu. Bogatov, A.A. Tyugashev // Proceedings of International Symposium for Reliability and Quality. – 2013. – Vol. 1. – P. 307-308.

Acknowledgments

A.A. Tyugashev wishing to acknowledge Victor Soifer and Anatoly Kalentyev for inspiration to be involved to spacecraft onboard software engineering and opportunity to continue researches after coming back from USA.