Ways of Providing Intelligent Consistent Real-Time Control for Cyber-Technical Systems

A.A. Tyugashev[1], S.P. Orlov[2]

[1]Samara State Transport University, Svobody street 2V, Samara, Russia, 443066
[2]Samara State Technical University, Molodogvardeyskaya street 244, Samara, Russia, 443100

Abstract. Railroad Transportation, Automated Manufactories, Nuclear Power Plants, Aerospace Missions can be reviewed as examples of a complex technical system with computer-based control. The subsystems of these systems, in turn, also contain dozens of devices, sensors, actuators, etc. We can name this phenomenon as a system of the systems. The important issue of the control of such a system is functioning in real-time. To achieve the system's goals we need to implement the specific schedule of the physically coordinated processes. The wrong synchronization of the processes executed by the equipment can lead to a catastrophe. Processes to be executed consume various resources with limited available levels. The overconsuming of each kind of resources leads to inconsistency. The notion of consistent control logic considering these issues is being defined in the paper. The set of real-time control algorithms must implement this control logic to complete required tasks in the changing environment even in case of some faults of the equipment. The paper analyses the aspects of complexity and requirements for 'intelligence' to provide consistent control. Also, the paper presents some ways of synthesis and verification of such intelligent and consistent control and presents developed software tools.

1. Introduction

Nowadays, Humanity uses a lot of very complex technical systems. Railroad Transportation, Automated Manufactories, Nuclear Power Plants, Spacecrafts [1,2] can be reviewed as examples of such systems. What do we suppose when talking about the 'complex' system? First, we expect the complex structure with a hierarchical organization – the system consists of dozens of subsystems, and, in turn, each of the subsystems contains various devices, sensors, actuators, etc. Second, the 'complex' system demonstrates complex behavior in a changing environment. How can we provide named systems with 'consistent' or 'right' control with consideration of these kinds of complexity and what the 'consistence' does mean in this case?

Each system designed to execute some useful tasks: transport passengers and goods from one geographical location to another, generate electric power, produce output, etc. Hereby, the system has a set of goals (tasks). For the moving objects (but not just for them only) such as trains, planes, cars the goals should be achieved at a specified moment of time, i.e. there are deadlines for the goals. Frequently, if we want to have some desired condition of the system at a specified moment of time, we need previously complete some preparatory procedures. Other procedures, post-actions, must be fulfilled after reaching the main goal (for example, cleaning of the cabin). Hence, the other very important feature of the considered systems is a Real-Time nature of functioning. In fact, in many cases, the system must implement not just an abstract 'tasks', but the timed sequences of logically coordinated and physically mutually dependent actions at pre-defined moments of time. Some of the actions have non-zero duration, so we must describe processes continued in time. In other words, considered systems have an active nature with some plans or schedules to be implemented. In Aerospace Industry, such plan usually called as ‘cyclogram’. In many practical situations, there are physically and logically founded restrictions not just for the sequence of the processes to be executed, but for synchronization of begins and ends of them. Some processes must have no overlaps in time, and the reasons for this issue could be very serious. These requirements could be formulated using the language of Real-Time Control Logic [3,4].

The additional aspect of the complexity of the control logic is caused by the possibility of some unpredictable events which might require change/adapt system’s plans to provide flexible reaction. The system must successfully complete the plan both in normal operations and in case of abnormal situations. In a picture reflecting the cyclogram, the fact that this particular process has to be executed in a specific situation only (for example, if some event happened) can be shown by color. Also, we can see the duration specified for the processes continued in time.

The intelligent consistent control supposes dependability and flexibility. As it was stated above, in case of some emergencies caused by faults of the equipment, the system goals should be achieved anyway. This is possible due to the redundancy of the equipment/apparatus. The designers of the complex system provide structural and functional redundancy in several ways. First, the duplication is widely used for critical mechanisms and aggregates. If some particular device will be crashed, the control system should detect this abnormal situation and switch to backup one. In other words, there is a very important ability of intelligent cybernetic systems - reconfiguration. Another successfully applied [2,5] way to parry the device’s failures is to utilize functional redundancy to use another subsystem in an abnormal situation. To do this, the control algorithms must ‘understand’ functional abilities of the various kinds of installed equipment and existence of the opportunities to use another unit to execute some task instead of initially intended for this purpose.

Usually, the control subsystem of the modern complex technical system uses computers running a special sort of software – control software. In Aerospace Industry this software called ‘flight control software’. This software issue commands to the onboard equipment coded as the sequence of electric impulses. Command can means, for example, «Activate the device 2 of the system 1 now» or «Switch the gyrodyne 2 off». The software name itself means this is a ‘soft’ entity having the appropriate level of flexibility to reconfigure the onboard apparatus to keep enough level of all kinds of the required functionality during the whole mission [2,6].

Of course, the system works under the influence of the environment. The required execution of the system’s plans is being dependent on external factors. On the other hand, the system’s outputs and activity change the external environment due to physical engagements (perhaps, with some time delay). We can state the existence of the mutual influence between the system and its environment. This specificity caused the following requirement for the system. The control means should provide control that can guarantee safety during the completion of the pre-defined set of tasks the complex technical system was built to execute.

The safety, in this case, means not only internal safety, i.e. keeping the devices and subsystems in serviceable ‘healthy’ conditions, but also the external safety. We mean that the system has its own influence on the external world, and we must keep various kinds of influence in defined borders. Moving objects should not damage the humans or arbitrary external entity. The emissions of the enterprise must be within the specified limits, and so on. The other side of this problem connected with the accurate consumption of the available resources during the functioning. Each device requires particular resources, for instance, electric power. Numerous devices can be turned to various regimes with different levels of consumption of the resources. The control rules of the named technical systems are being implemented by ‘control logic’.

The very important issue is the necessity of presence in control means of the complex technical system of some internal ‘reflection’ of the following aspects. First, we need a picture of the external environment and its factors we should take into account when implementing our plans. Second, we should have the image of the current condition of the controlled system itself with the means to describe the level of functionality/workability of our devices. And finally, we must have the representation of the goals with the understandings which ones are already done and which are waiting to be executed at which moments of time. In this context, we can apply the Ashby's cybernetical Law of Requisite Variety [7]: only a variety of control means can absorb a variety of controlled complex system and its behavior. Or: the control system's complexity (both hardware and software) reflects the complexity of the controlled system itself.

Hereby, we need to define the models for adequate describing the presented complex systems with the corresponding representation of the real-time control logic considering the requirements and restrictions stated above, and to find the methods for building this consistent control logic in practice.

2. The Method

We can underline the necessity of the following essential features for real-time intelligently and consistently controlled complex systems:

- Presence of internal reflection of the external environment, the image of the current condition of the system including information about the actual level of functional abilities of the installed devices – 'image of itself', and the knowledge about the plan (schedule) including data about already completed tasks and goals to be achieved in future.
- The ability of flexible self-reconfiguration based on an evaluation of the current situation and tasks to be executed
- System's control logic based on the real-time rules which might be flexibly updated and expanded.

Druzhinin and Kntorov [8] mentioned the levels of complexity of cybernetical systems.

- Deterministic S_1 systems with the rigid transformation rule input X into output Y
- Stochastic S_2 systems with the notable influence of random factors to results
- S_3 systems without well-defined rules of transformation input into output
- S_4 systems implementing the plans and achieving the pre-settled goals
- S_0 systems with choosing its own goals and changing the structure and adaptive reaction for the inputs

Using this approach, the considered systems might be classified as S_0 systems. The reasons are the following. First, we have the flexible control logic taking into account the different situations implemented by control software. Second, we can state the presence of the possibility of self-restructuration. And finally, the set of goals to be achieved might be updated during the operations.

How we can describe the real-time control logic used by these systems? When we are talking about the logic, we suppose the usage of axioms and rules. Naturally, we should utilize some reasoning based on the rules of logic. What can we review as the 'control logic' of the complex technical system? Rules can be formed as 'IF {antecedents/assumptions} THEN {conclusions}. For complex technical systems working in real-time mode, the best results could be provided by the timed versions of these rules, which can be specified in the following manner:

$$\alpha_1(t_{u1})\wedge\neg\,\alpha_2(t_{u2})\wedge\ldots\ \alpha_M(t_{uM})\rightarrow A_1(t_{a1})\wedge A_2(t_{a1})\wedge\ldots\ A_N(t_{aN}) \qquad (1)$$

There are logical variables (with the values TRUE and FALSE) on the left side of the formulae, and the actions on the right side. Some of the actions set or clear the logical conditions, so after the application of some rule, the truth of particular conditions can be changed. The very important aspect of the complex system interacting with the external environment by the physical processes is changing the conditions reflecting the current situation, in time. As we presented above in (1), we have the conjunction of the conditions (some with the logical negation) on the left side of the rule. It is possible to specify several rules with the same left part, so these rules can be used as connected by logical OR (disjunction). Consequently, in accordance with the logical completeness of DNF/CNF form of logical rules, we can declare the universalism of this approach for the description of any real-time control logic.

The problem of the synthesis of the consistent control logic requires performing the following transitions. Since we have the goals to be achieved by the system with the correspondent deadlines, we can then make the transition to the required schedule (set of the schedules for various scenarios depending on course of events) of the actions (processes). Each action requires some specific functionality. For instance, moving objects need some abilities in navigation and some abilities in communications. Meanwhile, navigation can be performed using GPS/GLONASS satellite's signals or using the inertial navigation system. A power supply is another kind of required functionality which can be provided by different devices, for example by the batteries or by solar panels. So, we can realize the transition from the process schedule to schedule of necessary functionality. Then we should make a transition from the functionality to the devices needed to provide it. At this moment, we have the schedule (again, schedules for various scenarios) of the work of the system's' devices. The next transition is the transition from this schedule to the set of rules of control logic formulated as (1). And then we can implement (by manual coding or by automated code generation, see []) this logic implemented in the control software. The reverse engineering problem is the problem of verification whether the logic implemented in control software corresponds to the goals and their deadlines. It supposes the transition from the existing software modules back to control logic's rules. We can use special procedures for the extraction of the control logic rules from the program code by analyzing the software modules, then for the restoration of the aforementioned schedules, and then for checking if the required goals are being achieved in time.

Consequently, the 'consistency' of the control logic means:

- Correspondence to the set of the required conditions of synchronization, for example, $f_1 << f_2$, f_3 *CH* f_5, f_1->f_5->f_7, prohibition of the intersection of particular processes f_{11} <> f_8 (it can be caused by the physical reasons, for instance, if the spacecraft's solar panel can shade the lens of Earth Remote Sensing instrument)
- Functioning without violation of the limits of available resources and allowed emissions
- Dependability, i.e. the completion of the set of required tasks should be guaranteed regardless of device failures and happening of the unforeseen situations.

Whenever we have the schedule built starting from the system's goals or extracted from the control programs, its compliance with synchronization requirements can be verified using the physical sense of the operators << (precedence in time), <> (prohibition of the overlapping), CH (begin-begin link), CK (end-end link), → (direct following), see the publications [3,8].

Further, we can define the complex technical system as the following tuple:

$$\{BA, G, CL, RS, CA, CS\} \qquad (2)$$

Where BA is the set of the devices with the correspondent set of their working modes; G is a set of goals to be implemented accompanied by the deadline for each goal; FS is a set of the kinds of functionality; CL – control logic presented as a set of timed rules; RS is the set of resources/emissions having an impact on consistent functioning of the system with the specified maximum allowed levels of consumption/emission; SC is the set of the constraints for right synchronization of the system's processes in the above- presented form.

Actually, the BA can be reviewed as the algebraic system [3] with the relation of belonging the device to a system, and there are relations between the devices and their working modes, and between the working modes, levels of provided functionality, resources and emissions. The restrictions for the minimal required level of each kind of functionality and maximal available levels of each kind of the resources are the other essential constraint for the consistent control logic along with the time restrictions.

To solve the problem we can use computer simulation in a special software tool to calculate the consuming of the resources and emissions for all mission time duration. Another simulation mechanism can allow checking whether the levels of all kinds of required functionality will be enough for achieving the system's goals even in case of arising of abnormal situations.

The problem of verification of the control logic is checking whether 1) the specified set of rules implements the schedule which guarantees to achieve the goals with compliance of their deadlines; 2) available/allowed levels of resources and emissions are not violated, and 3) existing restrictions SC are not violated. In case of the abnormal situation caused by the fault of a particular device, the control subsystem should check the level of degradation of the corresponding functionality, and then issue a special command to activate the appropriate substitution. These rules must be a significant subset of the real-time control logic rules.

3. Conclusions and Future Work

The mathematical model for an adequate description of the control logic for a complex technical system working in real-time mode has been defined in the article. We have considered two fundamental problems connected with the consistent control logic: the problem of its verification, and the problem of synthesis of consistent real-time control means.

As we declared above, the authors supervise the development of the prototype of a special software tool allowing verifying the existing control logic implemented in the source code of the control programs. To solve the problem of synthesis the logic with compliance to conditions of consistency, we are trying to utilize the power of modern Satisfiability Model Theories Solvers, see [9]. Another approach can be based on constraint programming. Previously, we had the experience with the use of logic programming to describe the real-time control algorithms and its restrictions [3,10].

4. Acknowledgments

Authors wish to acknowledge his partners from Samara Space Centre and JSC Information Satellite Systems for the many-years collaboration in the field of automated control of the spacecraft and the founder of this area of researches, Anatoly Kalentyev. The special thanks should be addressed to academician Victor Soifer granted the author the opportunity to continue this researches.

5. 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. CALS technology in the lifecycle of complex control programs / A.A. Kalentyev, A.A. Tyugashev – Samara: Scientific Center of Russian Academy of Sciences, 2006. – 266 p.

[4] Tyugashev, A.A. Language and Toolset for Visual Construction of Programs for Intelligent Autonomous Spacecraft Control // IFAC – PapersOnLine, 2016.

[5] Tyugashev, A.A. Structure and algorithms of the motion control system's software of the small spacecraft / A.A. Tyugashev, A.V. Filatov, I.S. Tkachenko, E.V. Sopchenko // CEUR Workshop Proc. – 2015. – Vol. 1490. – P. 246-251.

[6] Sygurov, Yu.M. Method for modeling of Spacecraft onboard apparatus and building of consistent control logic with limited onboard resources / Yu.M. Sygurov, A.A. Tyugashev // Journal of Physics Conference Series. – 2019. – Vol. 1368. – P. 042032.

[7] Ashby, W. Ross Introduction to Cybernetics // New York: Chapman & Hall, 1956.

[8] Druzhinin, V.B. The problems of the systemology (the problems of the theory of complex systems) // V.B. Druginin, D.S. Kontorov – Moscow: Sovetskoye Radio, 1976.

[9] Tyugashev, A.A. Application of SMT solvers for evaluation of Real-Time control logic of spacecraft // Journal of Physics: Conference Series. – 2019. – Vol. 1096. – P. 012156.

[10] Tyugashev, A.A. Visual Builder of Rules for Spacecraft Onboard Real-Time Knowledge Base // 8th KES International Conference on Intelligent Decision Technologies (KES-IDT). – 2016. – Vol. II. – P. 189-205.