Improving Dependability of Reconfigurable Robotic Control System

Eduard Melnik[1], Iakov Korovin[2], and Anna Klimenko[2(✉)]

[1] Southern Scientific Center of the Russian Academy of Sciences (SSC RAS), Rostov-on-Don, Russia

[2] Scientific Research Institute of Multiprocessor Computing Systems, Taganrog, Russia

anna_klimenko@mail.ru

Abstract. The robotic information and control systems based on performance redundancy and decentralized dispatching concepts are fault-tolerant and have a high potential in the dependability aspect. The paper deals with a dependability improvement through the configuration forming. As the load balancing improves the reliability function, the quality criteria of system configurations relates to load balancing. Configuration forming is a multicriteria and multiconstraint problem. The proposed approach is to replace monitoring and control tasks relocation criterion by delegating of task context distribution to the software components of the system. The paper contains a new simplified model of the configuration forming problem, the dependability improvement approach and simulation results being discussed briefly.

Keywords: Robotics · Mechatronic objects · Information and control system · Reconfiguration · Configuration forming · Reliability · Dependability

1 Introduction

Dependability of the robotic information and control system (ICS) is an important contemporary issue, because of wide usage of robotics and complex mechatronic objects nowadays. Robotics is a part of oil and gas production industry, chemical industries, power plants, spacecraft and aircraft, and, its dependability level is crucial very frequently. The failures of such complexes may impact the environment or lead to casualties, so a huge number of efforts has been made to improve the dependability level since the 1940s.

According to the definition, dependability is the ability to deliver service that can justifiably be trusted [1–3]. The service delivered by a system is its behavior as it is perceived by its user. The function of a system is what the system is intended to do, and is described by the functional specification. The service is correct when the system function is implemented. A system failure is an event which occurs when the delivered service deviates from the correct one.

Reliability is one of the attributes of dependability and relates to the service continuity. Practically, the measure of reliability is a reliability function – the probability that an object will be functioning beyond the specified time [4]. So, the reliability function improving is a way to achieve the dependability of acceptable level.

A. Ronzhin et al. (Eds.): ICR 2017, LNAI 10459, pp. 144–152, 2017.
DOI: 10.1007/978-3-319-66471-2_16

Besides this, the fault tolerance must be taken into account as a means of obtaining dependability. In practice, fault-tolerance is implemented by the system redundancy: first studies relate to the works of von Neumann, Moore and Shannon [5, 6], and their successors, who developed theories of using redundancy to build reliable logic structures from less reliable components. Nowadays the structural redundancy is used almost everywhere, although it has some disadvantages, which are: system weight, cost, and resource utilization issues in general [7].

Within this paper the reconfigurable ICSs with performance redundancy and decentralized dispatching (for more detailed information see [8–10]) are under consideration. The current research contains the novel approach to dependability improvement through the particular way of configuration forming. As it will be shown below, the proposed method is based on an assumption that with reducing the number of objective functions the solution quality is improved in terms of other objective functions. As the objective function presence is mandatory for the configuration forming problem, we can deliver its implementation by the third-party facilities, which also will be described precisely.

The current paper also contains an improved formal model of the configuration forming problem, which is clarified in comparison with [11, 12], a dependability improvement method description, some simulation results and discussion.

2 Reconfigurable ICS with Performance Redundancy

The main difference between structural and performance reservation approaches is that structural redundancy proposes the presence of reserve elements within the system, and performance redundancy operates with the elements with additional performance. The principles of performance redundancy are described more precisely in [10, 12]. The concept of the performance redundancy is the cause of the reconfiguration procedure design: the system deals with the redistribution of computational tasks from the faulted computational node to the operational ones. While ICS operates in the circumstances of real time, the term “configuration” becomes one of the key terms of the system.

In the scope of this paper the configuration is the way to distribute the monitoring and control tasks (MCTs) among the computational units (CU) of ICS. Such resource allocation must be implemented according to the data exchange constraints between MCTs and other constraints which are given by hardware and software system implementation.

In case of CU failure its MCTs must be launched on other CUs, and, besides, those MCTs can be allocated on more than one CU.

Such reconfiguration scheme relates to the system reliability function: the one’s value depends on CU temperature, which grows with loading increasing:

$$P_{CU} = P_{CU0}^{2^{k_d \cdot D/10}}, \tag{1}$$

where P_{CU} – reliability function value of loaded CU, P_{CU0} – reliability function value of CU without loading, k_d – temperature dependency on loading ratio, D – CU loading.

So, the spreading of MCTs between CUs with load balancing affects the system reliability, and load-balancing criteria should be included into the multicriteria objective function.

As was mentioned above, configurations are the key term of the chosen class of ICSs.

Decentralized monitoring and control of the ICS is implemented by multiagent system. Each agent is associated with its own CU, and each agent on the initialization stage of the ICS has an access to the list of possible system configurations, which are formed on the design stage of the system.

Configuration forming problem is discussed in detail in [11, 12], but it seems to be a little bit cluttered. The new simplified version of generalized configuration problem forming model will be presented below, and then the approach of dependability improvement will be presented.

3 Configuration Forming Problem

There are N MCTs with computational complexities g_i, M CUs with equal performance m_j, $U = \{u_{ij}\}$ – the percentage of j CU performance allocated for the i MCT, T – planned completion time for the N MCTs, $F = \{f_k\}$, $k \in \{1, \ldots M\}$, – the set of simultaneously failed CUs.

Through the resource allocation every MCT links to the CU, and it can be described by the following tuple:

$a_i = <j, u_{ij}, t_i>$, where j – the CU identificator, u_{ij} – the allocated resource ratio, t_i – the time of MCT i accomplishment.

So, the set $A = \{a_i\}$ determines the configuration of ICS before failure, the set $A' = \{ a_i'\}$ determines the configuration of ICS after the reconfiguration. In fact, A' is the solution of configuration forming problem, and a_i' – the tuples which describes the new MCT assignments.

The objective functions are as follows.

Firstly, the number of MCTs relocated from the operational nodes must be minimized. In other words, if there is a solution where the MCT's new assignment propose the relocation of tasks from the operational nodes, we should choose the solution, where the number of such relocations is as small as possible. This objective function can be described with the expressions given below.

Let's determine the subtraction operator for sets A and A' so that:

$$a_i - a_i' = \begin{cases} 0, \; if\, j\, is\ equal\ to\ j'; \\ \quad 1, \; otherwise. \end{cases} \tag{2}$$

Then:

$$F_1 = \sum_{i=1}^{N} (a_i - a_i') \rightarrow MIN. \tag{3}$$

The optimal location in the search space of this objective function means that only MCTs from the faulted node are relocated.

The second objective function is the minimization of the eliminated MCTs. In fact, some MCTs are critical and must be saved during the reconfiguration, and some MCTs are non-critical. But from the system survivability point of view it is extremely preferable to save as much MCTs as possible. So,

$$F_2 = |A| - |A'| \rightarrow MIN. \tag{4}$$

And, finally, the dispersion of CU loadings must be minimized:

$$F_3 = \sum_{k=1}^{K} u'_{kj} - \sum_{l=1}^{L} u'_{lq} \rightarrow MIN, \ \forall j, q, \tag{5}$$

where K is the number of MCTs assigned to the CU j, L is the number of MCTs assigned to the CU q.

The main constraint is that all MCTs must be accomplished within the planned completion time T:

$$t'_i + \frac{g_i}{u'_{ij} \cdot m_j} \leq T, \ \forall i, j. \tag{6}$$

Also the failed CUs must be taken into consideration:

$$M' = M - F, \tag{7}$$

where M’, M and F are the sets of CUs.

And, lastly, the bordering conditions are: all values of the variables are positive,

$$u_{ij} < 1, u'_{ij} < 1, \forall i, j.$$

At first glance the problem is similar to the k-partition problem, which has a suitable solving method, but vector objective function makes the problem np-hard with complex and non-trivial search space. Also it must be mentioned that with the increasing of objective function number the quality of solution degrades.

As the preferable attribute of the system is the load balancing, the goal of the configuration forming is to get solutions with as good load balancing as possible. At the same time the other objective functions must be taken into consideration.

It must be mentioned that Service Oriented Architecture (SOA) concept is used in contemporary ICSs, too. Services can be relocated, hence for the SOA-based ICSs there is no need to keep the MCT relocation criteria at all.

The next section contains the approach of dependability improvement description.

4 A Dependability Improvement Approach

The reliability function is one of the dependability attributes, so, with reliability improvement we increase the dependability level. Load balancing affects onto the CU reliability, hence, the solutions of the configuration problem forming should be as good as possible in terms of load balancing.

The configuration forming problem is a three-criterion in our particular case, but, perhaps, if at least one criterion is eliminated, the quality of solutions can be improved.

But the minimization of relocated tasks from the operational nodes is expedient because of MCT context data, which, in case of reconfiguration, must be transferred to the new assigned node through communicational network, and it can take unacceptable time and resources.

The concept of the approach presented is to delegate the MCT context data distribution to the CU agents and to design system configurations without criteria of MCT relocation, while these MCTs are located on the nodes, where they can be launched.

So, when the configurations are obtained, the agents form the list of data distribution and through the regular mode of ICS prepare the actual context for the possible task relocations. As a result, we have a kind of distributed data storage, which needs some additional algorithms for its functioning.

For instance, the steps described below can be done for the context data distribution.

ICS Initialization.

- CU agent searches the list of configurations for the CU ids, where current CU MCTs can be relocated in case of other node failure. The list of distribution is formed.
- If the list of distribution is not empty, the “intention” messages are sent.
- The confirmation messages are received.
- If the “intention” message was received, the confirmation message is sent to appropriate agent.

ICS Regular Mode.

- An agent takes the MCT context data and multicasts it according to the list of distribution.
- An agent receives the context data.

ICS Reconfiguration.

- The agent loads the new configuration.
- Search the list of distribution in order to deliver actual context data to the MCTs, which has become active on the current node.
- Well-timed data delivery.

5 Simulation Results and Discussion

For the simulation a random set of 25 MCTs with computational complexity 10–40 conventional units was generated. MCTs were assigned to the 10 CUs with equal performance. The cases of failures are combinations of one random failure and two random failures simultaneously. The criterion of the solution quality is the load balancing, because of its impact onto the reliability function.

Solutions were got with the simulated annealing technique (in details see [9]). It must be mentioned that the algorithm adaptation used gives the local optimums of the problem. To evaluate the quality of solutions, the equal number of SA iterations was used for both simulations.

The simulation results are given below. On Fig. 1 the maximum CU loadings are shown, Figs. 2, 3, 4, 5 and 6 are the detailed examples of the fruitful usage of the method considered.

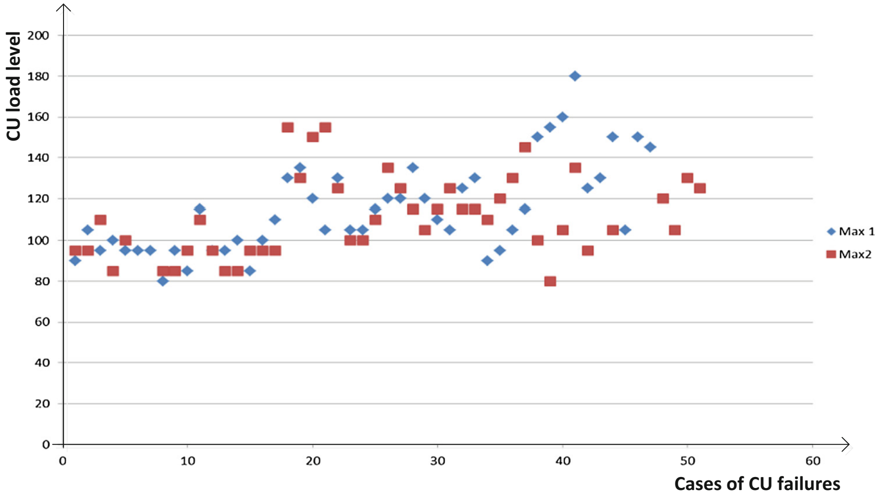

Fig. 1. Maximum CU loadings. The x-axis contains cases of CU failures, y-axis contains the loading values. Max 1 – solutions with objective functions (3, 4, 5). Max 2 – solutions with objective functions (4, 5)

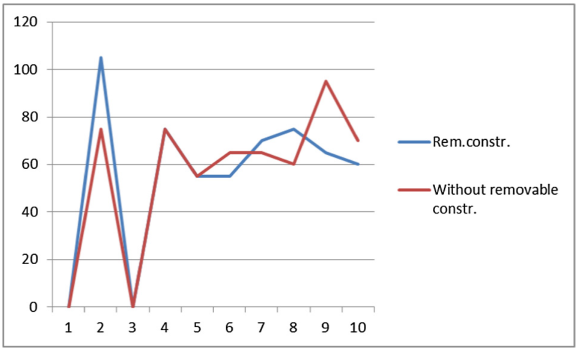

Fig. 2. CU load level when CU1 and CU3 are failed

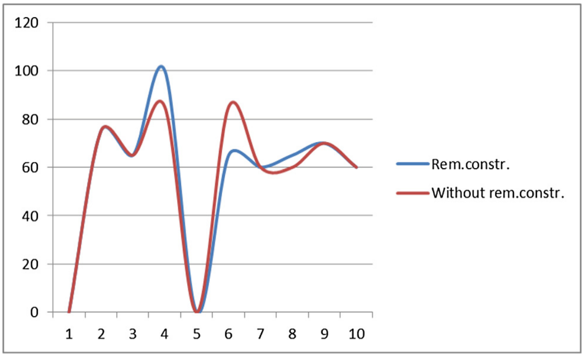

Fig. 3. CU load level when CU1 and CU5 are failed

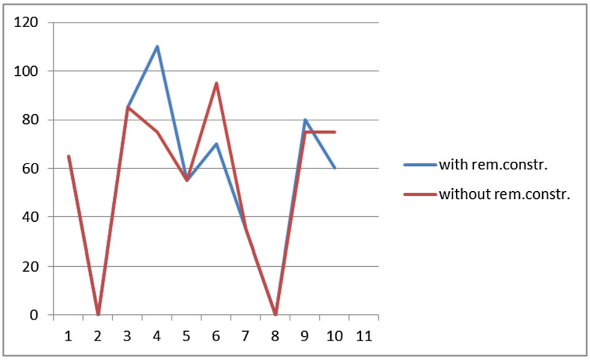

Fig. 4. CU load level when CU2 and CU8 are failed

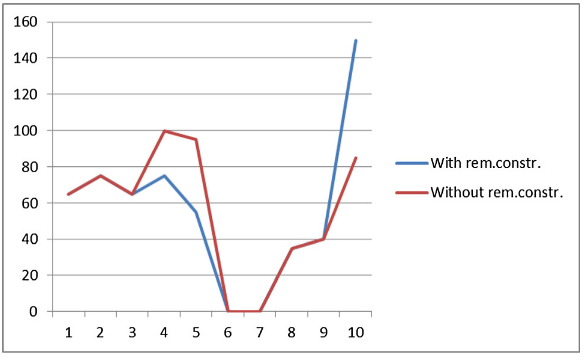

Fig. 5. CU load level when CU6 and CU7 are failed

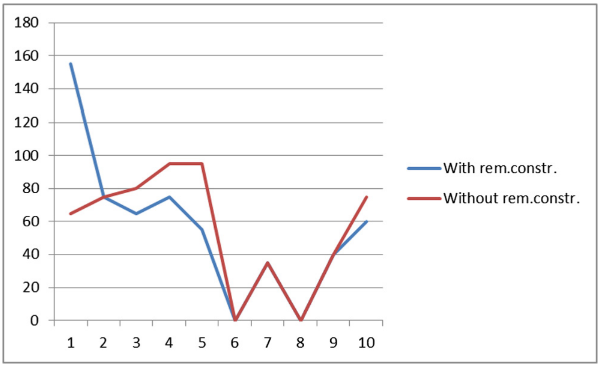

Fig. 6. CU load level when CU6 and CU8 are failed

There is a tendency on Fig. 1 of the difference of maximum loading dispersion growth with the growth of failed CU number. When the failed CU number equals 1, the maximum loadings are rather of the same magnitude. When the number of failed CUs is 2, the difference between solutions with all the criteria and without MCT relocation criterion is more obvious. It is seen that maximum loadings can be decreased, but, in some cases, the criterion removal does not produce any improvements. The probable reason of such behavior is the stochastic search particularity: with the fast, “quenching” temperature schemes the local (not global) optimums are found.

Figures 2, 3, 4, 5 and 6 contain the examples of load balancing with and without MCT relocation criterion. The cases of CU failures are shown on X-axis, and the Y-axis is the CU load level.

It is obvious that some load pikes are smoothen in the circumstances of equal SA iterations number, and we suppose that further, more precise simulations will confirm the revealed tendency.

At the same time, Figs. 1, 2, 3, 4, 5 and 6 allow to confirm, that it is expediently not to use only parallel simulated annealing search to improve the quality of solutions, but, besides this, make a search through the criteria elimination with assumption, that the criteria semantics can be delegated to the software (hardware) component of the system.

6 Conclusions

Within the scope of this paper the approach of the reconfigurable robotic control system dependability improvement was presented, described and discussed. The cornerstone of this approach is to get rid of MCT relocation criteria in multicriteria configuration forming problem and to delegate the semantic of the removed criteria to the software component of the system. Besides this, a new model of the configuration forming problem is given, simulation is done and analyzed briefly. According to the current stage of our study, it is expedient to form configurations not only with parallel search techniques (which allow to choose the best local minima), but also with the approach of “delegated” criterion. As an example, such “delegating” approach improves the solution quality up to the ratio of 1,7 (Fig. 6).

Acknowledgements. The reported study was funded by SSC RAS projects 02562014-0008, 0256-2015-0082 within the task 007-01114-16 PR and by RFBR project 17-08-01605-a.

References

1. Special session. Fundamental concepts of fault tolerance. In: Digest of FTCS-12, pp. 3–38 (1982)
2. Laprie, J.C.: Dependable computing and fault tolerance: concepts and terminology. In: Digest of FTCS-15, pp. 2–11 (1985)
3. Laprie, J.C.: Basic Concepts and Terminology. Springer, Heidelberg (1992)
4. Pham, H.: System Software Reliability. Springer Series in Reliability Engineering, 440 p. Springer, London (2006)
5. Von Neumann, J.: Probabilistic logics and the synthesis of reliable organisms from unreliable components. In: Shannon, C.E., McCarthy, J. (eds.) Annals of Math Studies, vol. 34, pp. 43–98. Princeton University Press, Princeton (1956)
6. Moore, E.F., Shannon, C.E.: Reliable circuits using less reliable relays. J. Franklin Inst. **262** (191–208), 281–297 (1956)
7. Zhang, Y., Jiang, J.: Bibliographical review on reconfigurable fault-tolerant control systems. Ann. Rev. Control **32**(2), 229–252 (2008)
8. Melnik, E., Korobkin, V., Klimenko, A.: System reconfiguration using multiagent cooperative principles. In: Abraham, A., Kovalev, S., Tarassov, V., Snášel, V. (eds.) First International Scientific Conference "Intelligent Information Technologies for Industry" (IITI 2016). Advances in Intelligent Systems and Computing, vol. 451, pp. 385–394. Springer, Heidelberg (2016). doi:10.1007/978-3-319-33816-3_38
9. Klimenko, A., Klimenko, V., Melnik, E.: The parallel simulated annealing-based reconfiguration algorithm for the real time distributed control fault-tolerance providing. In: 9 IEEE Application of Information and Communication Technologies, pp. 277–280 (2015)
10. Melnik, E.V., Korovin, I.S., Klimenko A.B.: A novel approach to fault tolerant information and control system design. In: 5-th International Conference on Informatics, Electronics Vision, University of Dhaka, Dhaka, Bangladesh (2016)
11. Korovin, I., Melnik, E., Klimenko, A.: A recovery method for the robotic decentralized control system with performance redundancy. In: Ronzhin, A., Rigoll, G., Meshcheryakov, R. (eds.) ICR 2016. LNCS, vol. 9812, pp. 9–17. Springer, Cham (2016). doi:10.1007/978-3-319-43955-6_2
12. Melnik, E.V., Klimenko, A.B.: Informational and control system configuration generation problem with load-balancing optimization. In: 10 IEEE Application of Information and Communication Technologies, pp. 492–496 (2016)