

УДК 004.424

doi: 10.21685/2072-3059-2025-1-5

Метод выбора рационального состава функционального программного обеспечения бортовых вычислительных систем робототехнических комплексов в экстремальных условиях

К. А. Суминов

Институт электронных управляющих машин имени И. С. Брука, Москва, Россия

konstantin.a.suminov@mcst.ru

Аннотация. *Актуальность и цели.* Ключевую роль в функционировании робототехнических комплексов (РТК) играют бортовые вычислительные системы (БВС), обеспечивающие переход от дистанционного управления к полу- и полностью автономным системам. Такой переход требует повышения сложности и адаптивности аппаратных и программных компонентов. Ввиду большого разнообразия решений с аппаратной, алгоритмической, архитектурной и других сторон, а также ограниченности вычислительных ресурсов БВС РТК выбор рационального состава функционального программного обеспечения при реконфигурации в процессе функционирования является нетривиальной задачей. *Материалы и методы.* Представлен метод выбора рационального состава функционального программного обеспечения БВС РТК в экстремальных условиях, который позволяет выбрать рациональную конфигурацию системы в зависимости от требований к решению поставленных задач исходя из текущего состояния внешней среды, внутреннего состояния БВС РТК и имеющихся ограничений. *Результаты и выводы.* Метод работает над многоверсионными библиотеками функциональных программ БВС РТК. Метод использует решение обобщенной задачи о мультипликативном многомерном рюкзаке с мультивыбором и дополнительными ограничениями. Такой подход позволяет учитывать сложные зависимости между элементами функционального программного обеспечения.

Ключевые слова: многоверсионное программирование, альтернативные решения, реконфигурация, бортовые вычислительные системы, робототехнические комплексы, БВС РТК, БВС Эльбрус, рациональный состав функционального программного обеспечения

Для цитирования: Суминов К. А. Метод выбора рационального состава функционального программного обеспечения бортовых вычислительных систем робототехнических комплексов в экстремальных условиях // Известия высших учебных заведений. Поволжский регион. Технические науки. 2025. № 1. С. 55–66. doi: 10.21685/2072-3059-2025-1-5

A method for selecting the rational composition of functional software for onboard computing systems of robotic complexes under extreme conditions

K.A. Suminov

Institute of Electronic Control Machines named after I.S. Bruk, Moscow, Russia

konstantin.a.suminov@mcst.ru

Abstract. *Background.* On-board computing systems (OCS) play a key role in the functioning of robotic complexes (RC), ensuring the transition from remote control to semi- and

fully autonomous systems. This transition requires increased complexity and adaptability of hardware and software components. Due to the wide variety of solutions from the hardware, algorithmic, architectural and other aspects, as well as the limited computing resources of the UAV RTK, the choice of a rational composition of functional software during reconfiguration during operation is a non-trivial task. *Materials and methods.* A method for selecting a rational composition of the functional software of the UAV RTK in extreme conditions is presented, which allows selecting a rational configuration of the system depending on the requirements for solving the assigned tasks based on the current state of the external environment, the internal state of the UAV RTK and the existing limitations. *Results and conclusions.* The method works on multi-version libraries of functional programs of the UAV RTK. The method uses the solution of the generalized problem of a multiplicative multidimensional knapsack with multichoice and additional constraints. This approach allows taking into account complex dependencies between elements of functional software.

Keywords: multiversion programming, alternative solutions, reconfiguration, onboard computing systems, robotic complexes, UAV RTK, Elbrus microprocessors, Rational composition of functional software

For citation: Suminov K.A. A method for selecting the rational composition of functional software for onboard computing systems of robotic complexes under extreme conditions. *Izvestiya vysshikh uchebnykh zavedeniy. Povolzhskiy region. Tekhnicheskie nauki = University proceedings. Volga region. Engineering sciences.* 2025;(1):55–66. (In Russ.). doi: 10.21685/2072-3059-2025-1-5

Введение

Развитие робототехники стремительно прогрессирует, что обусловлено повсеместной востребованностью робототехнических комплексов (РТК) в различных прикладных задачах. Современная техника всех видов, включая РТК, особенно активно развивается в области бортовых вычислительных систем (БВС). Для решения актуальных задач РТК становятся все более самостоятельными, переходя от дистанционного управления к полу- и полностью автономным системам [1], что неизбежно приводит к усложнению всех их компонентов [2].

БВС РТК часто состоят из множества вычислителей [3], которые могут отличаться своим предназначением (графический вычислитель, тензорный вычислитель и др.), архитектурой, вычислительными способностями.

При построении РТК, кроме большого множества альтернативных решений, с аппаратной стороны существует большое разнообразие программных реализаций алгоритмов, используемых для решения задач, стоящих перед РТК. Кроме статических, существуют динамические аспекты, возникающие в ходе эксплуатации РТК. Так, в процессе эксплуатации могут меняться приоритеты самих задач и требования к качеству их выполнения (повышение точности при прицеливании и т.д.). Кроме того, могут меняться внешние по отношению к РТК условия – сменилась погода, и внутренние – выход из строя вычислителя, перегрев узла и т.д.

Это порождает огромный выбор альтернативных решений, и возникает задача как определения исходной конфигурации, так и поиска наиболее рациональной конфигурации в сложившихся условиях при большом количестве параметров в процессе функционирования РТК. При этом в качестве критерия рациональности может выступать, например, живучесть РТК [4]. Обеспечение возможности решения такой задачи достигается созданием множества альтернативных решений для задач РТК с использованием многоверсионного программирования [5].

Примером могут служить ситуации, возникающие в применении РТК наземного РТК класса. На этапе создания, в зависимости от его массогабаритных характеристик, ставящихся требований и общих условий эксплуатации, необходимо определить аппаратную конфигурацию БВС. После начала выполнения поставленного задания (например, добраться с грузом из пункта А в пункт Б) могут возникать нештатные ситуации, в том числе ввиду преднамеренного воздействия. Так, при штатном движении может использоваться режим скрытности, при котором необходимо в соответствии с приоритетом выбрать программную компоновку, обеспечивающую наилучший показатель качества распознавания. При возникновении ситуации обнаружения РТК переходит в режим побега, при котором необходимо в соответствии с приоритетом выбрать компоновку функционального программного обеспечения (ФПО), обеспечивающую наилучший показатель скорости движения. При возникновении ситуации отказа одного из вычислителей БВС необходимо выполнить перераспределение решаемых задач между оставшимися вычислителями, максимизировав показатель качества в соответствии с приоритетом и т.д.

При переходе в каждое следующее состояние, характеризующееся своим приоритетом, необходимо менять конфигурацию для достижения наилучших показателей, одновременно с этим необходимо поддерживать минимально допустимые требования по всем характеристикам. То есть нельзя выбрать компоновку, обеспечивающую максимальную скорость, при которой качество распознавания будет неприемлемым. Данная задача является легко решаемой при наличии неограниченных ресурсов, но в реальных ситуациях ввиду ограничений необходимо осуществлять выбор конфигурации и состава ФПО БВС РТК. Сам же выбор строится на основе метода выбора рационального состава ФПО БВС РТК в экстремальных условиях, строящегося на использовании паспортов альтернативных решений, в полной мере опираясь на математический аппарат системного анализа [6].

Метод выбора рационального состава функционального программного обеспечения бортовых вычислительных систем робототехнических комплексов в экстремальных условиях

Метод выбора рационального состава работает над многоверсионными библиотеками функциональных программ БВС РТК и может быть представлен следующими шагами:

- сформировать многоверсионные библиотеки ФПО БВС РТК;
- провести предварительную фильтрацию решений из многоверсионной библиотеки функциональных программ;
- сформировать математическую модель решателя по описанным требованиям;
- программно реализовать математическую модель решателя;
- применить выполненную реализацию над полученными результатами фильтрации;
- обработать результаты.

Рассмотрим подробно каждый из пунктов, а также выполним краткое описание метода формирования многоверсионных библиотек функциональных программ БВС РТК.

Метод формирования многоверсионных библиотек функциональных программ бортовых вычислительных систем робототехнических комплексов

Построение множества альтернативных решений и их паспортов состоит из следующих этапов:

- определение перечня задач, ставящихся перед РТК;
- реализация множества альтернативных решений;
- определение перечня параметров и зависимостей для реализованных частных решений;
- проведение экспериментальных исследований с целью определения характеристик разработанных частных решений;
- построение паспортов моделей.

• *Определение перечня задач, ставящихся перед РТК*, – целевая задача, должна быть декомпозирована на конечное число подзадач. Каждая такая подзадача, в свою очередь, может включать конечное число более мелких частей и т.д. При этом стоит отметить, что такие элементы функционируют не изолированно друг от друга, а взаимодействуют как система, в которой свойства одного зависят как от внешних сложившихся условий, так и от условий, определяемых поведением других элементов.

• *Реализация множества альтернативных решений* – для каждой задачи T_i разработать множество альтернативных решений S_i :

$$S_i = \{S_{i1}, S_{i2}, \dots, S_{iC_i}\}, \quad (1)$$

где S_{ij} – частное решение задачи T_i ; C_i – количество решений для i -й задачи. Альтернативные решения могут различаться по следующим аспектам: алгоритмические подходы, языки программирования, программные библиотеки и их версии, архитектурные решения.

• *Определение перечня параметров и зависимостей для реализованных частных решений* – чтобы осуществлять выбор между различными алгоритмами, необходимо иметь параметры, по которым их сравнивать. В данном случае рассматриваются два типа параметров: категориальные и функциональные [7].

Категориальные параметры могут принимать значение из определенного множества значений. Примером таких параметров может служить характеристика алгоритма, на каком вычислителе он может быть исполнен (E2K, X86) или версия необходимой библиотеки для исполнения данного алгоритма (OpenCV Version ≥ 3.2).

Функциональные параметры представляют собой функцию, описывающую потребление ресурса, например памяти, в зависимости от параметров алгоритма.

• *Проведение экспериментальных исследований* с целью определения характеристик разработанных частных решений путем статистической обработки экспериментальных данных результатов моделирования.

• *Построение паспортов моделей* – на основе проведенных экспериментальных исследований и определенных характеристик решений необходимо составить некоторый «паспорт модели». Пример таблицы с такими «паспортами» для M задач в общем виде представлен в табл. 1.

Таблица 1

Пример таблицы с паспортами решений для М задач

Задача	1					2	...	М
Номер параметра	Тип	Решение 1	Решение 2	...			...	...
1	a	$f_{111}(\overline{x_{111}})$	$f_{121}(\overline{x_{121}})$	...	$f_{1C_11}(\overline{x_{1C_11}})$	...	...	$f_{MC_{M1}}(\overline{x_{MC_{M1}}})$
...	...	...	...	...	...	...	...	...
j	a	$f_{11z}(\overline{x_{11z}})$	$f_{12z}(\overline{x_{12z}})$	...	$f_{1C_1z}(\overline{x_{1C_1z}})$	...	...	$f_{MC_{Mz}}(\overline{x_{MC_{Mz}}})$
$j+1$	b	x_{111}	x_{121}	...	x_{1C_11}	...	...	$x_{MC_{M1}}$
...	...	...	...	...	...	...	...	...
N	b	x_{11N-z}	x_{12N-z}	...	x_{1C_1N-z}	...	...	$x_{MC_{MN-z}}$

Предварительная фильтрация решений

Основной целью применения предварительной фильтрации является **обеспечение соответствия текущим условиям**, т.е. отбор решений, которые могут быть применены в заданных внешних и внутренних условиях с целью **сокращения пространства поиска**. Исключение альтернатив, которые противоречат ограничениям РТК, позволяет получить **снижение вычислительных затрат**, что непосредственно влечет уменьшение времени для последующих этапов.

Фильтрация выполняется на основе анализа многоверсионной библиотеки функциональных программ и построения матрицы подходящих решений в соответствии с вектором текущего состояния БВС и РТК.

Формирование математической модели решателя

В основе выбора решения, удовлетворяющего ресурсам, оптимального по какому-либо критерию, лежит обобщенная задача о мультипликативном многомерном рюкзаке с мультивыбором [8] и дополнительными ограничениями, что каждый класс может быть представлен только в одном рюкзаке, и из каждого класса можно взять не более одного предмета. Кроме того, следует учесть, что некоторые решения могут зависеть от решений другого класса задач, а также требуют решения задач, являющихся для них вспомогательными, но не представляющими ценности в обособленном виде. То есть при решении необходимо рассматривать вспомогательные решения и учитывать топологические зависимости между решениями. Рассмотрим формулировку задачи в контексте имеющейся задачи выбора рационального состава ФПО БВС РТК.

Пусть имеется:

- Множество вспомогательных решений $\{d_1, d_2, \dots, d_t\}$, где t – количество вспомогательных решений. Каждое вспомогательное решение d_i , $i=1, 2, \dots, t$, имеет: m различных характеристик потребления ресурсов;

$w_{ij} \geq 0$ – значение потребления ресурса j -го типа для решения i , $j = 1, 2, \dots, m$; вектор состояния Sd_i длиной P , содержащий необходимые зависимости по библиотекам, архитектуре вычислителя и др.

• Множество основных решений $\{r_1, r_2, \dots, r_n\}$, где n – количество основных решений, каждое решение r_i , $i = 1, 2, \dots, n$, имеет: критерий качества $v_i \geq 0$, m различных характеристик потребления ресурсов, $w'_{ij} \geq 0$ – значение потребления ресурса j -го типа для решения i , $j = 1, 2, \dots, m$. Также для каждого i -го основного решения определено множество зависимостей $Dr_i \subseteq \{r_1, r_2, \dots, r_n\} \setminus \{r_i\}$ от других основных решений и $Dr_i \subseteq \{d_1, d_2, \dots, d_v\}$, от вспомогательных задач, вектор состояния Sr_i длиной P , содержащий необходимые зависимости по библиотекам, архитектуре вычислителя и др.

Все n основных решений разбиты на k классов задач $G_1, G_2, \dots, G_k$. Каждое решение может принадлежать только одному классу, т.е. множества G_a , $a = 1, 2, \dots, k$, являются непересекающимися,

Также имеется x вычислителей, каждый вычислитель имеет: ограничение на объем по потребляемому ресурсу, w''_{ij} – ограничение на ресурс j -го типа для вычислителя i , $j = 1, 2, \dots, m$, $i = 1, 2, \dots, x$; вектор состояния Sx_i длиной P , содержащий значения, определенные на данном вычислителе значения для зависимостей решений (версии программных библиотек на борту, архитектуры вычислителя и др.).

Все j типов ресурсов для вспомогательных решений, основных решений и имеющиеся в вычислителях совпадают между собой.

В таком случае необходимо найти набор основных и вспомогательных решений, распределенных по вычислителям с общим максимальным критерием качества, соблюдая при этом ограничение по суммарному потреблению ресурса каждого типа для каждого вычислителя и ограничение, что из каждого класса основных решений можно взять только одно решение и один раз, каждый класс задач может быть представлен только в одном вычислителе. Решения из множества вспомогательных решений рассматриваются как дополнения к зависимому от них основному решению и рассматриваются к обязательному размещению на вычислителе только при размещении соответствующего основного решения. Основное решение может быть размещено, только если все его зависимости также размещены к выполнению на вычислителях.

Таким образом, для применения рассматриваемого метода необходимо сформировать математическую модель решателя в соответствии с описанной постановкой задачи поиска и применить ее программную реализацию к отфильтрованному множеству решений, после чего обработать полученные наборы.

Практика и результаты

В рамках проводимых в Институте электронных управляющих машин имени И. С. Брука работ был реализован такой подход с использованием многоверсионного программирования для некоторых основных задач назем-

ных РТК. Эксперимент проводился для БВС, построенной на основе двух микропроцессоров серии «Эльбрус», а именно «Эльбрус-8С» и «Эльбрус-2С3» [9, 10]. На рис. 1 представлена структурная схема рассматриваемой БВС.

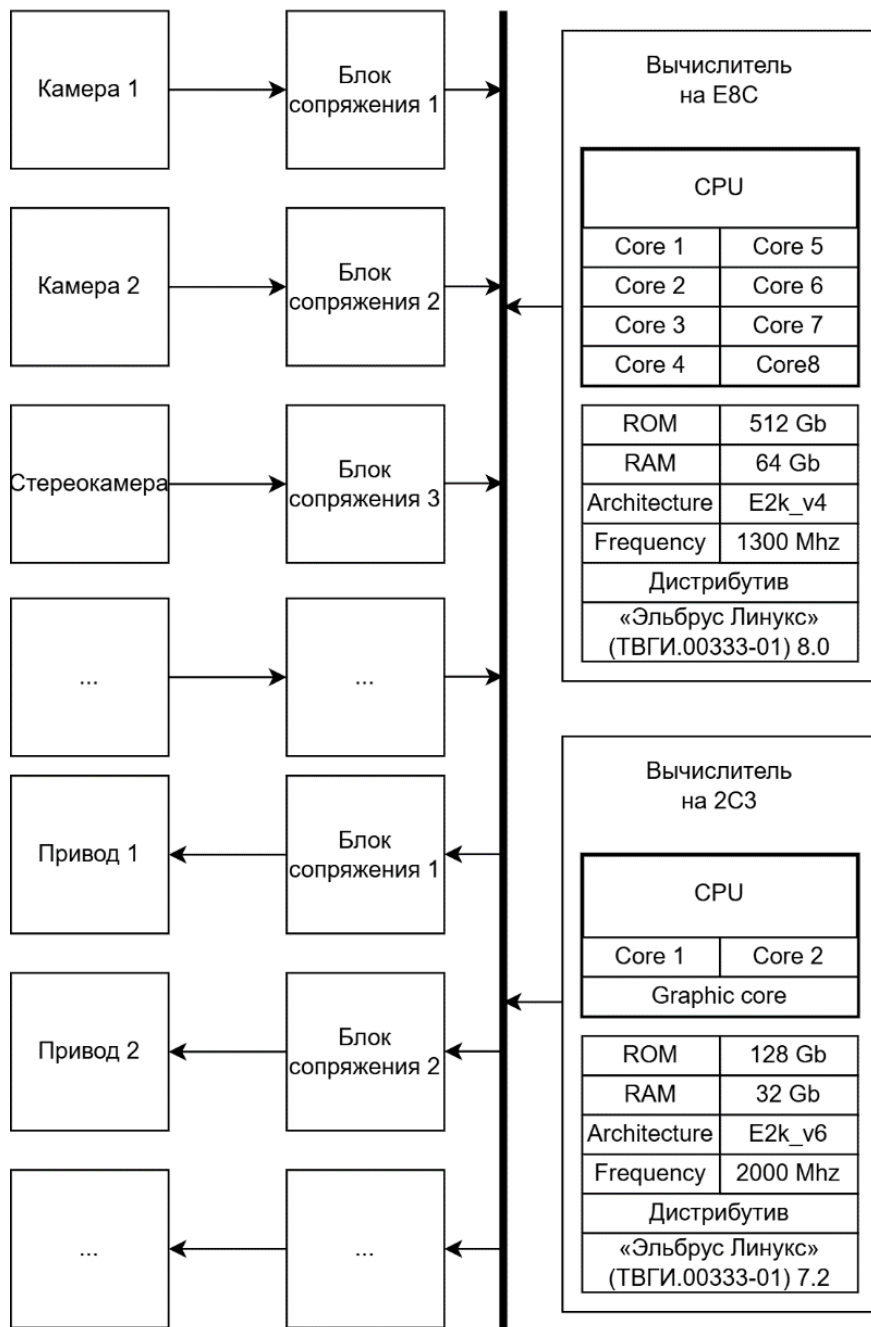


Рис. 1. Структурная схема рассматриваемой БВС

В результате построения множества альтернативных решений было получено 662 различных частных решений для 8 классов решаемых задач: задачи обнаружения объектов, анализа текстур, сегментации изображений, стерео-

реконструкции [11], распознавания объектов, поиска геометрических примитивов, конструирования графа проходимости, поиска пути на графе проходимости.

По результатам проведенных экспериментов для каждого частного решения был получен паспорт с его характеристиками, функциональными зависимостями. Каждое решение описывается 57 различными функциональными и категориальными параметрами.

Было проведено моделирование для двух сценариев применения наземного РТК на примере РТК класса тележка. Сценарий 1 представляет собой последовательность повторяющихся состояний, принимаемых РТК, которая соответствует режиму функционирования без изменения приоритетов для выполняемых задач. Сценарий 2 представляет собой последовательность различных состояний, принимаемых РТК, которая соответствует режиму функционирования с изменением приоритетов для выполняемых задач. Переход между состояниями в каждом из сценариев содержит вероятность отказа вычислителя БВС РТК.

С помощью разработанного метода и программных средств получены оценки увеличения живучести РТК с БВС на основе микропроцессоров отечественного производства серии «Эльбрус» от 1,2 до более чем в 20 раз. На рис. 2,*а* приведен график оценки живучести для сценария 1 с и без использования метода, их отношения показаны на рис. 2,*б* в зависимости от вероятности отказа (в процентах).

Видно, что при нулевой вероятности отказов при переходах между состояниями живучесть остается неизменной, что связано с постоянным приоритетом при функционировании РТК и неизменной конфигурацией БВС РТК. При повышении вероятности отказов при переходах живучесть с использованием разработанного метода становится выше, чем без его использования, что обусловлено тем, что при неизменных приоритетах меняется объем доступных ресурсов. При выходе из строя вычислительного комплекса (ВК) БВС РТК без изменения состава ФПО БВС РТК все задачи, выполняемые на вышедшем из строя вычислительном комплексе, становятся недоступны, при изменении же состава ФПО БВС РТК подбирается компоновка, покрывающая все задачи исходя из доступных ресурсов.

На рис. 3,*а* приведен график оценок живучести для сценария 2 в зависимости от вероятности отказа вычислительных узлов. Оценки приведены для случаев с использованием разработанных средств и без них. На рис. 3,*б* приведен график отношения указанных оценок в зависимости от вероятности отказа вычислительных узлов.

Видно, что при сценариях с изменяемыми приоритетами в процессе функционирования, даже при нулевой вероятности отказов, и соответственно, неизменной конфигурации БВС РТК, живучесть с использованием разработанного метода значительно выше, чем без его использования. Это обусловлено тем, что при исходной конфигурации учитывается лишь один стартовый приоритет РТК, с использованием же метода при каждом изменении состав ФПО БВС РТК выбирается так, чтобы максимизировать соответствующий текущему состоянию приоритет. При увеличении вероятности отказа разница становится еще более значительной (более чем 20 раз), что так же, как и для сценария 1, обусловлено выбором состава ФПО БВС РТК исходя из новой конфигурации вычислительного комплекса БВС.

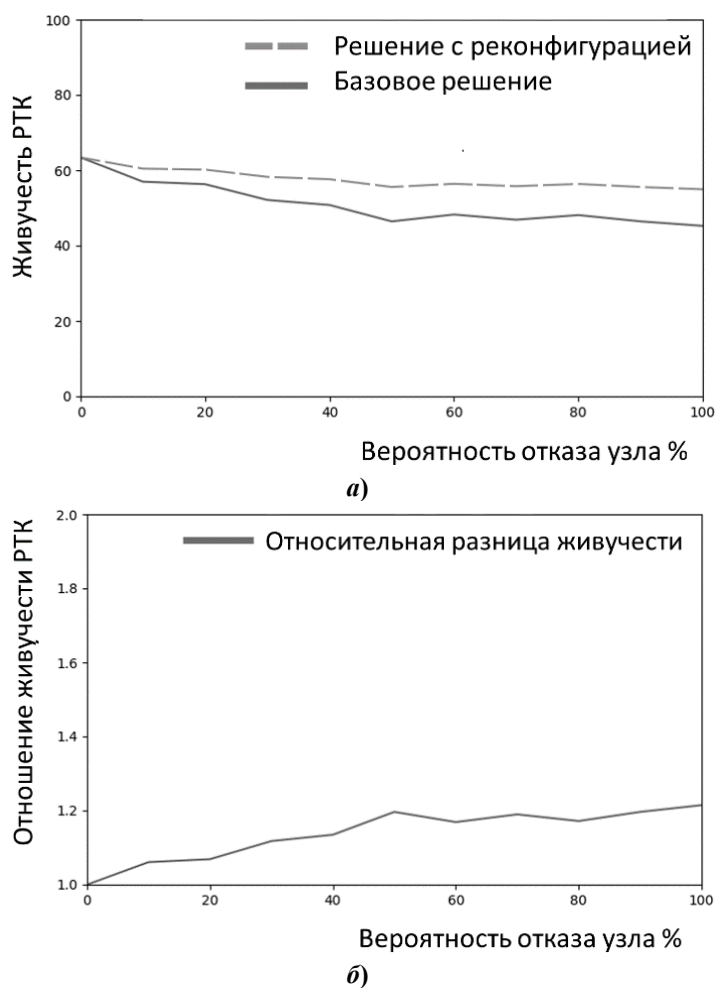


Рис. 2. Зависимость живучести РТК для сценария 1 с и без использования разработанного метода от вероятности отказа узла БВС (а), зависимость отношения живучести для сценария 1 с и без использования разработанного метода от вероятности отказа узла БВС (б)

Заключение

В результате разработан метод выбора рационального состава ФПО БВС РТК в экстремальных условиях. Предложен метод формирования многоверсионных библиотек функциональных программ БВС РТК и построение их паспортов. Рассмотрены аспекты предварительной фильтрации решений из многоверсионной библиотеки функциональных программ. Рассмотрено формирование математической модели решателя для выбора рационального состава ФПО БВС РТК.

Проведенные экспериментальные исследования с моделью обеспечения живучести БВС РТК на БВС, построенном на базе отечественных микропроцессоров «Эльбрус-2С3» и «Эльбрус-8С», показали, что в зависимости от сценариев применения живучесть может быть повышена более чем в 20 раз по отношению к решению без использования разработанного метода. Показано, что разница наиболее выражена при сценариях функционирования РТК

с изменчивыми приоритетами и высокой вероятностью отказа ВК из состава БВС РТК.

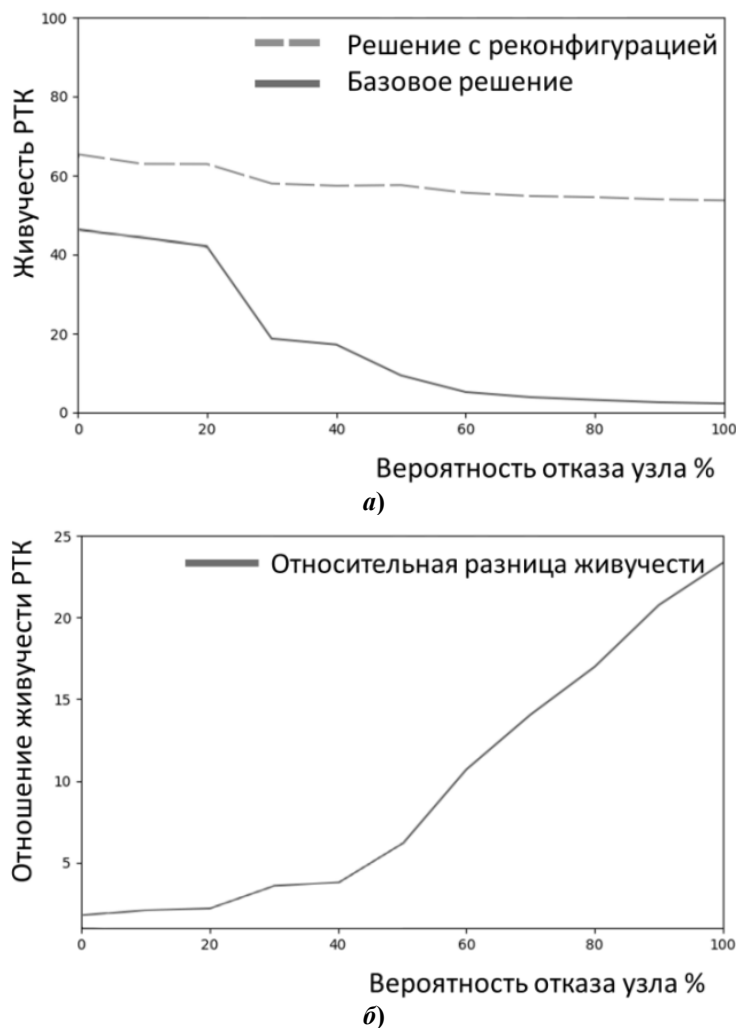


Рис. 3. Зависимость живучести РТК для сценария 2 с и без использования разработанного метода от вероятности отказа узла БВС (а), зависимость отношения живучести для сценария 2 с и без использования разработанного метода от вероятности отказа узла БВС (б)

Показано, что разработанный метод позволяет как синтезировать программы, оптимальные по выбранному критерию в зависимости от условий или архитектуры ВК, так и производить формирование рационального состава ФПО БВС РТК для обеспечения возможности реконфигурации в процессе функционирования.

Список литературы

1. Бочаров Н. А., Парамонов Н. Б., Александров А. В., Славин О. А. Решение задач когнитивного управления группой роботов на многоядерных микропроцессорах «Эльбрус» // Современные информационные технологии и ИТ-образование : сб.

- науч. тр. II Междунар. науч. конф. и XII Междунар. науч.-практ. конф. / под ред. В. А. Сухомлина (Москва, 24–26 ноября 2017 г.). М., 2017. С. 232–244. EDN: IUMMTK
2. Романов А. М. Обзор аппаратно-программного обеспечения систем управления роботов различного масштаба и назначения. Ч. 3. Экстремальная робототехника // Российский технологический журнал. 2020. Т. 8, № 3 (35). С. 14–32. doi: 10.32362/2500-316X-2020-8-3-14-32 EDN: VOELFM
 3. Lei Deng, Song Han, Yuan Xie [et al.]. Model compression and hardware acceleration for neural networks: A comprehensive survey // *Proceedings of the IEEE*. 2020. Vol. 108, № 4. P. 485–532.
 4. Калюжный А. В., Гончаренко В. А., Соколовский А. Н. Проблемы повышения живучести робототехнических систем космического назначения на основе много-агентных технологий // Вестник Российского нового университета. Серия: Сложные системы: модели, анализ и управление. 2020. № 1. С. 168–173. doi: 10.25586/RNU.V9187.20.01.P.168 EDN: FUGUSH
 5. Парамонов Н. Б., Минин И. В. Многоверсионное моделирование в ходе проектирования информационных систем // Межотраслевая информационная служба. 2014. № 1. С. 40–44.
 6. Перегудов Ф. И., Тарасенко Ф. П. Введение в системный анализ. М. : Высшая школа, 1989.
 7. Лаптев В. В., Морозов А. В. Экспертная система оценки качественных характеристик проектной модели программного обеспечения // Информационные технологии. 2009. № 3. С. 8–11.
 8. Martello S. Knapsack problems: Algorithms and computer implementations. New York : J. Wiley & Sons, 1990.
 9. Кожин А. С. Основные проектные решения для процессора «Эльбрус-16С» // Наноиндустрия. 2020. Т. 13, № S4. С. 74–75.
 10. Бычков И. Н., Лобанов И. Н., Молчанов И. А. Вычислительная техника на основе аппаратно-программной платформы «Эльбрус» для перспективных информационных систем // Приборы. 2018. № 8 (218). С. 14–20.
 11. Бочаров Н. А., Зуев А. Г., Славин О. А. Производительность микропроцессора Эльбрус-8СВ для решения задач технического зрения в условиях ограничений энергопотребления // Известия ЮФУ. Технические науки. 2021. № 1 (218). С. 259–271. doi: 10.18522/2311-3103-2021-1-259-271 EDN: PUHSYC

References

1. Bocharov N.A., Paramonov N.B., Aleksandrov A.V., Slavin O.A. Solving problems of cognitive control of a group of robots on multi-core microprocessors “Elbrus”. *Sovremennye informatsionnye tekhnologii i IT-obrazovanie: sb. nauch. tr. II Mezhdunar. nauch. konf. i XII Mezhdunar. nauch.-prakt. konf. / pod red. V.A. Sukhomlina (Moskva, 24–26 noyabrya 2017 g.) = Modern information technologies and IT education: proceedings of the 2nd International scientific conference and the 12th International scientific and practical conference / edited by V.A. Sukhomilin (Moscow, November 24–26, 2017)*. Moscow, 2017:232–244. (In Russ.). EDN: IUMMTK
2. Romanov A.M. Review of hardware and software for control systems of robots of various scales and purposes. Part 3. Extreme robotics. *Rossiyskiy tekhnologicheskiy zhurnal = Russian technology journal*. 2020;8(3):14–32. (In Russ.). doi: 10.32362/2500-316X-2020-8-3-14-32 EDN: VOELFM
3. Lei Deng, Song Han, Yuan Xie et al. Model compression and hardware acceleration for neural networks: A comprehensive survey. *Proceedings of the IEEE*. 2020;108(4):485–532.
4. Kalyuzhnyy A.V., Goncharenko V.A., Sokolovskiy A.N. Issues of increasing the survivability of robotic systems for space purposes based on multi-agent technologies.

- Vestnik Rossiyskogo novogo universiteta. Seriya: Slozhnye sistemy: modeli, analiz i upravlenie* = *Bulletin of Russian New University. Series: Complex systems: models, analysis and control*. 2020;(1):168–173. doi: 10.25586/RNU.V9187.20.01.P.168 EDN: FUGUSH
5. Paramonov N.B., Minin I.V. Multi-version modeling in the design of information systems. *Mezhotraslevaya informatsionnaya sluzhba* = *Inter-industry information service*. 2014;(1):40–44. (In Russ.)
 6. Peregudov F.I., Tarasenko F.P. *Vvedenie v sistemnyy analiz* = *Introduction to Systems Analysis*. Moscow: Vysshaya shkola, 1989. (In Russ.)
 7. Laptev V.V., Morozov A.V. Expert system for assessing the qualitative characteristics of a software design model. *Informatsionnye tekhnologii* = *Information technology*. 2009;(3):8–11. (In Russ.)
 8. Martello S. *Knapsack problems: Algorithms and computer implementations*. New York: J. Wiley & Sons, 1990.
 9. Kozhin A.S. Main design solutions for the Elbrus-16S processor. *Nanoindustriya* = *Nanoindustry*. 2020;13(S4):74–75. (In Russ.)
 10. Bychkov I.N., Lobanov I.N., Molchanov I.A. Computing technology based on the hardware and software platform “Elbrus” for advanced information systems. *Pribory = Devices*. 2018;(8):14–20. (In Russ.)
 11. Bocharov N.A., Zuev A.G., Slavin O.A. Elbrus-8SV microprocessor performance for solving machine vision problems under energy consumption limitations. *Izvestiya YuFU. Tekhnicheskie nauki* = *Proceedings of South Federal University. Engineering sciences*. 2021;(1):259–271. (In Russ.). doi: 10.18522/2311-3103-2021-1-259-271 EDN: PUHSYC

Информация об авторах / Information about the authors

Константин Александрович Суминов
начальник отдела 3.4.7, Институт
электронных управляющих машин
имени И. С. Брука (Россия, г. Москва,
ул. Вавилова, 24)
E-mail: konstantin.a.suminov@mcst.ru

Konstantin A. Suminov
Head of the department 3.4.7, Institute
of Electronic Control Machines named
after I.S. Bruk (24 Vavilova street,
Moscow, Russia)

Автор заявляет об отсутствии конфликта интересов / The author declares no conflicts of interests.

Поступила в редакцию / Received 12.12.2024

Поступила после рецензирования и доработки / Revised 03.02.2025

Принята к публикации / Accepted 24.02.2025