

Яцук Г.Е., Уманский А.Б., Ананьин А.С.

АО «НПО автоматики им. академика Н.А. Семихатова», Екатеринбург, Россия

ОЦЕНКА КАЧЕСТВА ОТРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ СИСТЕМ АВТОМАТИЧЕСКОГО УПРАВЛЕНИЯ НА МОДЕЛИРУЮЩИХ И КОМПЛЕКСНЫХ СТЕНДАХ

В работе рассматривается проблема оценки надежности программного обеспечения (ПО) и эффективности отработочных позиций (стендов). Разработана методика на основе анализа вершин графа путей исполнения ПО, позволяющая учесть специфику позиции, на которой отрабатывается то или иное множество путей исполнения ПО. Учет особенностей и структуры ПО предлагается осуществлять с использованием представления ПО на основе графовой модели. Предложенная модель позволяет достоверно определить пути исполнения ПО на основе анализа параметров путей управляющего графа. Преимуществом модели являются: простота предлагаемого метода; учет структурной сложности ПО; возможность уточнения вероятности бессбойной работы (ВБР) ПО по результатам эксплуатации без проведения полной отработки ПО. Практической ценностью предлагаемой модели является возможность оценки параметра ВБР ПО без необходимости проведения испытаний в полном объеме.

Ключевые слова: надежность ПО, метрики, сложность ПО, графовая модель, отработочный стенд, эффективность.

Авторы заявляют об отсутствии возможных конфликтов интересов.

Для цитирования: Яцук Г.Е., Уманский А.Б., Ананьин А.С. Оценка качества отработки программного обеспечения для систем автоматического управления на моделирующих и комплексных стендах. Труды Крыловского государственного научного центра. 2018; специальный выпуск 1: 198–204.

УДК 621.4:681.3.06

DOI: 10.24937/2542-2324-2018-1-S-I-198-204

Yatsuk G., Umanskii A., Ananyin A.

Scientific and Production Association of Automatics named after academician N.A. Semikhatov, Yekaterinburg, Russia

SOFTWARE QUALITY ASSURANCE FOR AUTOMATIC CONTROL SYSTEM ON SIMULATION AND COMPLEX TESTBEDS

The software reliability evaluation and efficiency of testbeds are considered. A method based on analysis of software-execution path graph nodes is developed, which takes into account specifics of modeling a test bed used to examine one or another set of software execution paths. A graph model is suggested to take into account specifics of the software structure. The proposed model makes it possible to reliably determine the software paths by analysis of the control flow graph parameters. The merits of this model include simplicity of the method, consideration of sophisticated software structure, possibility to accurately estimate the probability of fail-free software operation based on service database without full testbedding of software. The practical advantage of the proposed model is that it is possible to assess fail-free software operation probabilities without full scope of testbedding.

Key words: software reliability, metrics, software sophistication, graph model, test bed, efficiency.

Authors declare lack of the possible conflicts of interests.

For citations: Yatsuk G., Umanskii A., Ananyin A. Software quality assurance for automatic control system on simulation and complex testbeds. Transactions of the Krylov State Research Centre. 2018; special issue 1: 198–204 (in Russian).

UDC 621.4:681.3.06

DOI: 10.24937/2542-2324-2018-1-S-I-198-204

Модель оценки надежности программного обеспечения

Model for evaluation of software reliability

Традиционно оценка показателей надежности ПО и их прогнозирование производится на базе математических моделей надежности программ. Надежность программы является характеристикой ее исполняемого кода. Обработка программы проводится для подтверждения выполнения требований технического задания на разработку системы управления (СУ). На начальных этапах разработки ПО ставится задача обеспечения функционирования по «положительной ветке», т.е. должно быть достигнуто прохождение всех режимов и задач при условии отсутствия неисправностей в приборах СУ. Как правило, прохождением по «положительным веткам» – самый простой этап. Наиболее сложным и емким процессом обработки является проверка функционирования по «отрицательным веткам», с внесением неисправностей или имитацией нештатных ситуаций.

Если представить алгоритм, в соответствии с которым разработано и функционирует ПО, в виде управляющего графа $G(V, E)$, то при функционировании по «положительной ветке» прохождение ПО осуществляется по одному и тому же пути L , т.е. вероятность бесспорного исполнения $P(L)=1$. С учетом того, что, например, для СУ летательного аппарата существует множество режимов (предстартовая подготовка, полет, зачетные или сдаточные режимы, режимы с имитацией полета), то для n -режимов при функционировании в штатных режимах при отсутствии неисправностей путем обработки подтверждается, что

$$i = 1 \div n, \forall i : P(L_i) \rightarrow 1,$$

где L_i – путь, пройденный по управляющему графу при функционировании ПО в i -м режиме работы; n – количество режимов.

Соответственно, при исполнении ПО по каждому пути L_i осуществляется прохождение через множество V_i вершин управляющего графа G . При этом множество вершин V_i для путей прохождения L_i пересекается, в этом случае можно выделить максимально достижимый подграф $G_0(V_0, E)$, причем такой, что $G_0(V_0, E) \in G(V_i, E), V_0 \in V_i$.

Иными словами, даже в процессе функционирования по различным штатным режимам производится обработка или подтверждение безотказности работы общей вершины, представленной подграфом $G_0(V_0, E)$.

Для этапов обработки на моделирующих и комплексных стендах (КС) при функционировании по «положительным веткам» обеспечивается полное покрытие условий и решений путей L_i , каждый из которых соответствует тому или иному режиму функционирования. Покрытие условий и решений зависит от возможностей комплексов для обработки СУ.

В режиме предстартовой подготовки СУ функционирует в проверочных режимах, поэтому вероятность выполнения предстартовой подготовки с учетом последовательного и независимого исполнения $i=1 \dots n$ задач может быть определена как

$$P_{\text{пн}} = \prod_i P(L_i),$$

где i – соответствующий номер проверочной задачи; $P(L_i)$ – вероятность исполнения i -задачи.

Для режима основной полетной программы вероятность исполнения равна

$$P_{\text{опп}} = P(L_{\text{опп}}),$$

т.е. исполнения пути $L_{\text{опп}}$ основного направленного управляющего графа.

При обработке на КС необходимо применение штатной аппаратуры, в качестве исполнительных элементов и датчиков используются их эквивалентные имитаторы. В силу штатного исполнения аппаратуры возможности позиции КС в части полноты покрытия могут быть ограничены возможностями применяемой имитирующей аппаратуры, которая подключается к штатным приборам, и математических моделей. Комплексная обработка ведется по принципу «черного ящика».

Обработка происходит последовательно, с постепенным увеличением количества обработанных путей исполнения ПО. В процессе обработки устанавливаются более приоритетные направления, определяемые в порядке очередности, исходя из следующих принципов:

- первоначально обрабатываются «положительные ветки» прохождения ПО;
- обработка основных «отрицательных веток», спровоцированных возникновением наиболее вероятных неисправностей постоянного характера в аппаратуре СУ, имитацию которых наиболее просто произвести в техническом плане;
- обработка основных «отрицательных веток», спровоцированных возникновением менее вероятных неисправностей аппаратуры СУ, которые носят сложный характер (кратковременный сбой, возникновение более одной неисправности), имитацию которых технически сложно выполнить.

Таким образом, на конкретном этапе обработки вероятность безотказного исполнения ПО равна вероятности безотказного исполнения множества путей L^m (m -весовой коэффициент), причем L^m входит в состав множества L :

$$L_i^m \in L^m; L^{m-1} \in L^m; L^m \in L.$$

Причем общее количество путей равно метрике Маккейба. Все предыдущие пути, состоящие в подмножестве с коэффициентом m , входят в состав подмножества с коэффициентом $(m+1)$. Весовой коэффициент m в этом случае должен быть целочисленным, что определяет первоочередность обработки. Чем коэффициент ниже, тем более приоритетным является этап обработки.

Таким образом, на начальных этапах обработки вероятность безотказного исполнения программы равна $P=P(L^1)$, т.е. множеству путей L_i^1 . Для более поздних этапов обработки, в процессе эксплуатации СУ, эта вероятность равна $P=P(L^m)$.

По итогам обработки программа считается правильной, если она не содержит ошибок. При этом можно описать среднее значение вероятности безотказного исполнения:

$$\bar{P} = \frac{1}{n} \cdot \sum_{i=1}^n P(L_i).$$

Формула дает возможность оценить качество ПО по результатам обработки.

Оценка надежности программы прошивки flash

Reliability of flash software

В качестве практического применения графовой модели предложен пример оценки надежности для ПО прошивки flash-памяти вычислителя бортовой цифровой вычислительной системы (БЦВС). На рис. 1 приведен вид алгоритма для задачи записи информации в flash-память вычислителя. Вид управляющего графа приведен на рис. 2. Данная задача используется при функционировании СУ автономного подвижного объекта и предназначена для обновления flash вычислителя.

Для приведенного на рис. 2 управляющего графа метрика Маккейба равна 11, т.е. граф имеет 11 путей исполнения. Положительным исполнением задачи являются пути с входящими в них вершинами:

L_1 : 1-2-3-4-5-7-8-9-10-11-12-15-16-18-19-20-21-7-13-23-24-25;

L_2 : 1-2-3-4-5-7-8-9-10-11-12-15-16-18-19-20-21-7-8-9-10-11-13-14-1-2-3-6-22-24-25.

Вероятность исполнения пути по «положительной» ветке будет равна произведению ВБР вершин V_i . Для вершин 1 и 25 по умолчанию $P=1$, т.к. они не являются значимыми.

Для пути L_1 вероятность исполнения определяется следующим образом:

$$P(L_1) = \prod_i P(V_i),$$

$i = 1 \div 12, 15, 16, 18, 19, 20, 21, 7, 13, 23, 24, 25.$

Обработка ПО проводилась в течение 9 этапов. На каждом этапе выполнялась доработка ПО вершин, расчет проводился по результатам исполнения ПО на целевой аппаратуре БЦВС. График значения $P(L)$ в зависимости от этапа обработки приведен на рис. 3.

Таким образом, для обеспечения возможности практического применения оценки надежности ПО необходимо:

- подготовить первоначальные данные, опираясь на результаты анализа структуры и содержимого ПО, а также исходные данные. Следует учитывать, что первоначальные данные должны содержать требования к объему тестирования и отражать структурную сложность ПО на основе метрики Маккейба, а также ее компонент (количество вершин графа);
- провести тестирование ПО с использованием подготовленных первоначальных данных;
- вести регистрацию всех результатов тестирования;
- выполнить расчеты на основе данных тестирования.

Методика оценки эффективности отработочных стендов

Methods for estimation of test bed efficiency

Как отмечалось выше, обработка ПО производится в несколько этапов и на разных позициях. Отработочные позиции могут иметь различные технические характеристики и возможности. Используя полученные результаты, возможно как провести оценку эффективности обработки для одной конкретной позиции, так и сравнить полученный результат для разных позиций.

В зависимости от позиции обработка проводится с использованием штатных приборов СУ, либо их эквивалентов с использованием технологической аппаратуры, которая позволяет:

- контролировать исполнение программ как в среде разработки, так и на целевой аппаратуре в режиме реального времени;
- считывать содержимое регистров, ОЗУ и ПЗУ в процессе исполнения программы;
- контролировать взаимодействие БЦВС с внешними подсистемами по требуемым интерфейсам.

Отработка может проводиться с использованием нескольких технических позиций, например:

- на рабочем месте программиста. Позиция позволяет проводить отладку программы без учета взаимодействия с внешними подсистемами БЦВС, а при необходимости – без учета межмодульного взаимодействия внутри БЦВС;
- в составе комплекса отработки аппаратуры и программ. Позиция позволяет проводить отработку с использованием взаимодействия БЦВС с внешними системами;
- в составе цифрового моделирующего комплекса, где проводится отработка алгоритмов функционирования БАСУ;
- в составе КС, где производится отработка с использованием штатной аппаратуры.

Та или иная позиция обладает характерными преимуществами и недостатками. Например, внутреннюю логику функционирования ПО без учета внешнего взаимодействия можно полностью отработать только на рабочем месте программиста. Отработку с учетом внесения неисправностей и кодо-

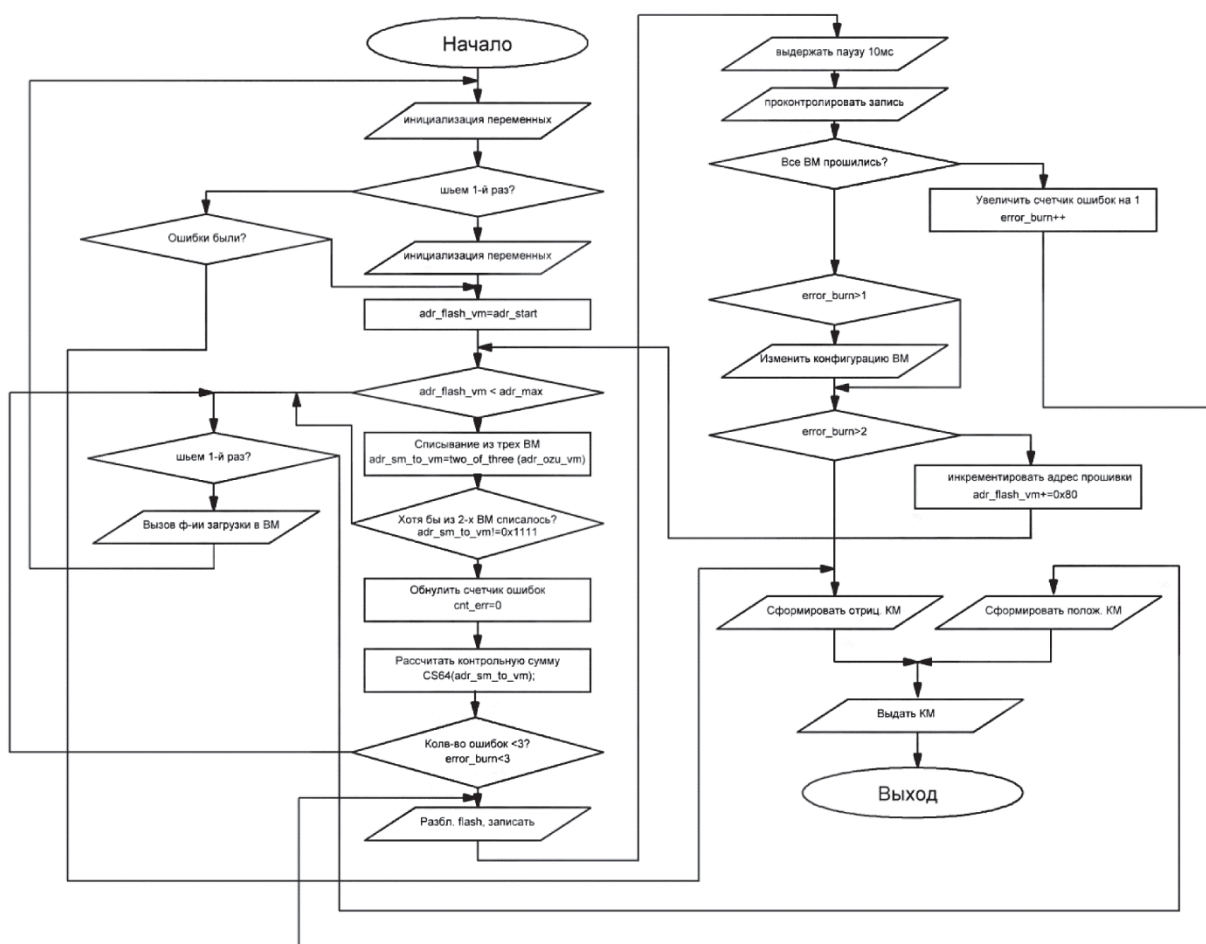


Рис. 1. Алгоритм задачи записи информации в flash-памяти вычислителя

Fig. 1. Algorithm for writing data into flash-memory of computer

вого взаимодействия между системами наиболее эффективно можно выполнить в составе комплекса отработки аппаратуры и программ. На КС возможно проведение отработки с учетом внесения наиболее вероятных и простых технических неисправностей в штатную аппаратуру. Поэтому на практике задача решается путем разделения задач, подлежащих отработке на той или иной позиции, а также задействования каждой позиции в необходимом объеме, что в итоге обеспечивает высокое качество отработки.

Предлагаемая модель надежности ПО позволяет оценить качество отработки ПО для отработочной позиции: без учета специфики отработочной позиции можно определить тот объем ПО, который можно отработать исходя из технических возможностей по отношению к требуемому объему. Для этого введем $K_{ко}$ – коэффициент качества отработки. Поскольку в процессе функционирования в общем виде априорной оценке подлежит конечный результат, то очевидной оценкой является определение соотношения количества путей исполнения ПО, которое возможно проверить на отработочной позиции к общему количеству:

$$K_{ко} = m/n, \quad (1)$$

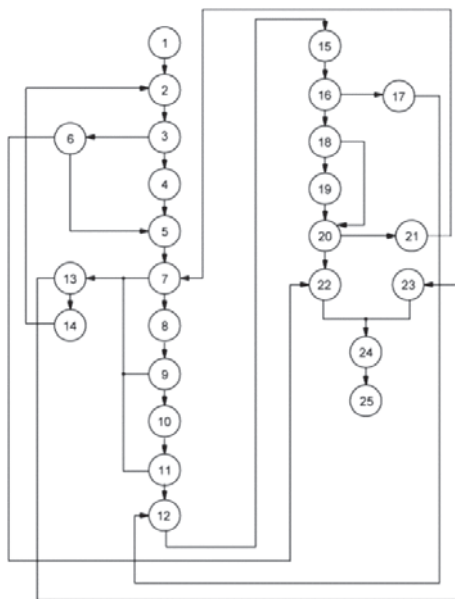


Рис. 2. Управляющий граф задачи записи информации в flash-памяти вычислителя

Fig. 2. Control flow graph for writing data into flash-memory of computer

где m – количество путей исполнения ПО, которое возможно проверить на отработочной позиции; n – общее количество исполнения путей ПО. Максимальное значение $K_{ко}$ не может превышать 1.

На практике на отработочных позициях невозможно целиком выполнить полностью тот или иной путь исполнения ПО. Например, на позиции рабочего места программиста невозможно отработать взаимодействие БЦВС с внешними системами. В этом случае расчет по формуле (1) затруднен.

Более точную оценку качества отработки можно получить, учитывая задействование вершин графа путей исполнения ПО. Для этого необходимо использовать соотношения количества вершин 1-го и 2-го рода, которые возможно проверить на отработочной позиции, к общему количеству вершин. При этом необходимо учесть, что:

- вершины 1-го рода имеют одно множество Y , связанное с передачей управления к последующим вершинам;
- вершины 2-го рода могут иметь несколько множеств Y , также связанных с передачей потока управления.

Для дальнейшего анализа следует использовать матрицу смежности. Введем α (коэффициент учета вершин 2-го рода) и β (коэффициент учета вершин 1-го рода).

Для вершин 2-го рода число выходящих ребер из i -й вершины 2-го рода в любую j -ю вершину равно 1, что характеризует отсутствие варианности по потоку управления графа. Коэффициент α учета вершин 2-го рода вычисляется следующим образом:

$$\alpha = \frac{m_{V^2}}{n_{V^2}},$$

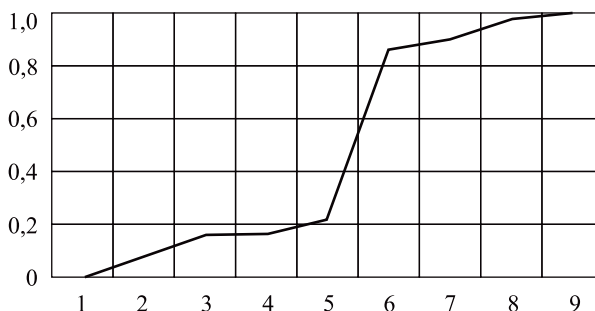


Рис. 3. Значение $P(L)$ в зависимости от этапа отработки

Fig. 3. $P(L)$ versus phase of software testbedding

где m_{V_2} – количество вершин 2-го рода, которое возможно проверить на данной отработочной позиции; n_{V_2} – количество вершин 2-го рода, которое необходимо проверить.

Для вершин 1-го рода число выходящих ребер из i -й вершины 1-го рода в любую j -ю вершину равно сумме элементов a_{ij} матрицы смежности, что характеризует наличие вариантности по потоку управления графа. Коэффициент β учета вершин 2-го рода вычисляется следующим образом:

$$\beta = \frac{\sum_i \sum_j a_{ij}^m}{\sum_i \sum_j a_{ij}^n}.$$

В этом случае $K_{\text{ко}}$ можно представить как

$$K_{\text{ко}} = \frac{m_{V_2} + \sum_i \sum_j a_{ij}^m}{n_{V_2} + \sum_i \sum_j a_{ij}^n}.$$

Эмпирически возможно определить относительное значение $K_{\text{ко}}$ для разных отработочных позиций. Однако на практике происходит декомпозиция процесса отработки, которая предполагает, что на каждой позиции упор делается на отработку тех или иных элементов ПО. С учетом применения нескольких отработочных позиций и того, что на каждой отработочной позиции обрабатывается определенное множество путей исполнения ПО, $K_{\text{ко}}$ определяется следующим образом:

$$K_{\text{ко}} = \sum_{i=1}^n K_{\text{ко}}^i,$$

где $i = 1 \dots n$ – количество отработочных позиций; $K_{\text{ко}}^i$ – коэффициент качества отработки для i -й позиции.

Таким образом, при организации процесса отработки стремятся получить $K_{\text{ко}} \rightarrow 1$.

Заключение

Conclusion

Учет особенностей и структуры ПО предлагается осуществлять с использованием представления ПО на основе графовой модели. Данная модель позволяет достоверно определить пути исполнения ПО на основе анализа параметров путей управляющего графа. Преимущества модели: простота предлагаемого метода; учет структурной сложности ПО; возможность уточнения ВБР ПО по результатам экс-

плуатации без проведения полной отработки. Практическая ценность предлагаемой модели заключается в возможности оценки параметра ВБР ПО без необходимости проведения испытаний в полном объеме. Необходимо отметить, что выполнение такого большого объема работ целесообразно проводить с использованием средств автоматизации с целью снижения вероятности внесения случайных ошибок в процессе подготовки тестирования. По результатам эксплуатации СУ необходимо проводить учет выявленных ошибок. Поскольку ошибки могут возникать при исполнении вершин V_i , то при обнаружении ошибки и ее устранении отработке подлежит только та часть ПО, которой соответствует вершина V_i . Соответственно, по мере устранения ошибок ВБР ПО будет расти и стремиться к 1.

Библиографический список

References

1. Яремчук С.А. Метод оценки количества программных дефектов с использованием метрик сложности // Радиоэлектронные и компьютерные системы. 2012. № 5(57). С. 212–218. [S. Yaremchuk. Method for estimating the number of software defects using software complexity metrics // Radioelectronic and computer systems. 2012; 5(57): 212–8. (In Russian)].
2. Черников Б.В., Поклонов Б.Е. Оценка качества программного обеспечения. М.: ИД «Форум» Инфра-М, 2012. [B. Chernikov, B. Poklonov. Software quality assessment. M.: ID “Forum” Infra-M, 2012. (In Russian)].
3. Ларионцева Е.А. Использование метрических характеристик программ при проведении сертификационных испытаний. Наука и образование. М.: МГТУ им. Н.Э. Баумана, 2012. [E. Lariontseva. Software metrics in certification tests. Nauka i obrazovanie. M.: BSTU, 2012. (In Russian)].
4. Пальчун Б.П., Юсупов Р.М. Оценка надежности программного обеспечения. СПб.: Наука, 1994. [B. Palchun, R. Usupov. Estimation of software reliability. SPb.: Nauka. 1994. (In Russian)].
5. Маевский Д.А., Яремчук С.А. Метод вероятностной оценки количества дефектов в программных модулях // Сборник Одесского политехнического университета. 2013. Вып. 1(40). С. 73–80. [D. Maevskiy, S. Yaremchuk. Method for probabilistic estimation of defects in software modules // Collection of papers of Odessa Polytechnic University. 2013; 1(40): 73–80. (In Russian)].
6. Липаев В.В. Тестирование компонентов и комплексов программ. М.: Синтег, 2011. [V. Lipaev. Testing of software components and packages. M.: Sinteg, 2011. (In Russian)].

7. *Луцаев В.В.* Проектирование и производство сложных программных продуктов. М.: Синтег, 2011. [*V. Lipaev.* Design and production of sophisticated software products. M. Sinteg, 2011. (In Russian)].
8. *Журавлев А.В., Антимиров В.М.* Методика оценки эффективности отработочной позиции СУ РКТ // Труды I Российско-белорусской научно-технической конференции «Элементная база отечественной радиоэлектроники». Н. Новгород: Нижегородское региональное отделение НТО радиотехники, электроники и связи им. А.С. Попова. 2013. № 4. С. 214–217. [*A. Zhuravlev, V. Antimirov.* Methods for workout tests on control systems of rocket and space technology. Proc. of I Russian-Belorussian Scientific & Technical Conference “Hardware components of home radio-electronics”. N. Novgorod: Regional Nizhy Novgorod branch of Radio Engineering, Electronics and Communication Scientific & Technical Society named after A.S. Popov. 2013; 4: 214–7. (In Russian)].
9. *Журавлев А.В., Шашмури И.В., Петухов В.И., Хохряков В.А.* Архитектура системы управления комплексом обработки аппаратуры и программ САУ РКТ // Ракетно-космическая техника. Сер. XI. Системы управления ракетных комплексов. 2013. Вып. 2. С. 16–26. [*A. Zhuravlev, I. Shashmurin V. Petukhov, V. Khokhryakov.* Architecture of control system for a complex of facilities intended for development control system hardware and software for rocket and space technologies // Rocket and Space Technologies. Ser. XI Rocket control systems. 2013; 2: 16–26. (In Russian)].

Сведения об авторах

Уманский Алексей Борисович, к.т.н, начальник сектора АО «Научно-производственное объединение автоматики им. академика Н.А. Семихатова». Адрес: 620075, Россия, г. Екатеринбург, ул. Мамина-Сибиряка, д. 145. Тел.: +7 (343) 263-76-89; E-mail: pdwn1982@yandex.ru.

Яцук Георгий Евгеньевич, начальник сектора АО «Научно-производственное объединение автоматики им. академика Н.А. Семихатова». Адрес: 620075, Россия, г. Екатеринбург, ул. Мамина-Сибиряка, д. 145. Телефон: +7 (343) 263-76-89; E-mail: rx9cim@rambler.ru.

Ананьин Александр Сергеевич, инженер-конструктор АО «Научно-производственное объединение автоматики им. академика Н.А. Семихатова». Адрес: 620075, Россия, г. Екатеринбург, ул. Мамина-Сибиряка, д. 145. Телефон: +7 (343) 263-76-89; E-mail: ananyinac@yandex.ru.

About the authors

Umanskii A., Candidate of Technical Sciences, Head of sector, Scientific and Production Association of Automatics named after academician N.A. Semikhatov. Address: ul. Mamina-Seberyaka 145, Yekaterinburg, 620075, Russia. Tel.: +7 (343) 263-76-89; E-mail: pdwn1982@yandex.ru.

Yatsuk G., Head of sector, Scientific and Production Association of Automatics named after academician N.A. Semikhatov. Address: ul. Mamina-Seberyaka 145, Yekaterinburg? 620075, Russia. Tel.: +7 (343) 263-76-89; E-mail: rx9cim@rambler.ru.

Ananyin A., Design engineer, Scientific and Production Association of Automatics named after academician N.A. Semikhatov. Address: ul. Mamina-Seberyaka 145, Yekaterinburg, 620075, Russia. Tel.: +7 (343) 263-76-89; E-mail: ananyinac@yandex.ru.

Поступила / Received: 14.02.18

Принята в печать / Accepted: 18.04.18

© Яцук Г.Е., Уманский А.Б., Ананьин А.С., 2018