

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И АВТОМАТИЗАЦИЯ НА ТРАНСПОРТЕ

DOI: 10.21821/2309-5180-2018-10-3-639-646

THE PRINCIPLES GOVERNING THE CONSTRUCTION OF ALGORITHMS AND DEBUG THEM FOR USE IN THE SYSTEMS OF VESSEL TRAFFIC MANAGEMENT

D. S. Vasilchenko

Admiral Makarov State University of Maritime and Inland Shipping,
St. Petersburg, Russian Federation

In this article, the principles of debugging control algorithms, the basis for the development of a subsystem for debugging algorithms for use in a complex control system, as well as the role of this subsystem in the structure of the ship traffic control system. The relevance of this issue is justified by the arising problems or the incorrect functioning of the software, which is part of the management information system. On the basis of a software module, the debugging subsystem can be implemented, being both an autonomously-functioning element and part of a control system with dependent communications. The issues under consideration play a significant role in the complex debugging of software of various sizes, and the proposed solution will help optimally use the resources of managing computer systems. As the object of integration, the ship traffic control system is considered, which contains certain groups of algorithms represented by software modules. The proposed approach is useful for diagnosing and reconfiguring the hardware and software of the control system. The development, testing and operation of software and hardware that use control algorithms must have not only practical significance, but also a logical-based order of execution. The use of the algorithm debugging subsystem can contribute both to the adaptation of the system to dynamically changing functions, and to the search for solutions in the event of errors in the operation of the system modules, which can sometimes lead to an incorrect functioning of the entire system. Adding a debugging subsystem cannot always be used as part of the basic functions of the system, since this process is rather an additional tool for modifying the operation, rather than a mandatory base component.

Keywords: control task, the stability of the computational process, the adaptation of control processes, the debugging of algorithms, the automation of the debugging process, the detection of algorithm errors, the technique of debugging algorithms, the description of algorithms, the hierarchical system, the modernizability of the system.

For citation:

Vasilchenko, Dmitrii S. "The principles governing the construction of algorithms and debug them for use in the systems of vessel traffic management." *Vestnik Gosudarstvennogo universiteta morskogo i rechnogo flota imeni admirala S. O. Makarova* 10.3 (2018): 639–646. DOI: 10.21821/2309-5180-2018-10-3-639-646.

УДК: 004.021

ПРИНЦИПЫ ПОСТРОЕНИЯ УПРАВЛЯЮЩИХ АЛГОРИТМОВ И ИХ ОТЛАДКИ ДЛЯ ИСПОЛЬЗОВАНИЯ В СИСТЕМАХ УПРАВЛЕНИЯ ДВИЖЕНИЕМ СУДОВ

Д. С. Васильченко

ФГБОУ ВО «ГУМРФ имени адмирала С. О. Макарова»,
Санкт-Петербург, Российская Федерация

В статье рассмотрены принципы отладки управляющих алгоритмов, основы для разработки подсистемы отладки алгоритмов для использования в комплексной управляющей системе, а также роль данной подсистемы в структуре системы управления движением судов. Актуальность данного вопроса обоснована возникающими неполадками или неверным функционированием программных средств, которые

являются частью управляющей информационной системы. На основе программного модуля подсистема отладки может быть реализована, являясь как автономно функционирующим элементом, так и частью управляющей системы с зависимыми связями. Рассматриваемые вопросы играют весомую роль при комплексной отладке программного обеспечения различного масштаба, и предложенное решение поможет оптимально использовать ресурсы управляющих компьютерных систем. В качестве объекта интеграции рассматривается система управления движения судов, которая содержит определённые группы алгоритмов, представленных программными модулями. Предлагаемый подход полезен для диагностики и реконфигурации программно-аппаратного обеспечения управляющей системы. Разработка, тестирование и эксплуатация программно-аппаратных средств, которые используют управляющие алгоритмы, должны иметь не только практическую значимость, но и логически-обоснованный порядок исполнения. Использование подсистемы отладки алгоритмов может способствовать как адаптации системы к динамически изменяющимся функциям, так и поиску решения в случаях возникновения ошибок в работе модулей системы, что иногда может привести к неверному функционированию всей системы. Добавление подсистемы отладки не всегда может использоваться в составе базовых функций системы, так как данный процесс является, скорее, дополнительным инструментом для модификации функционирования, чем обязательным базовым компонентом.

Ключевые слова: задача управления, устойчивость вычислительного процесса, адаптация процессов управления, отладка алгоритмов, автоматизация процесса отладки, обнаружение ошибок алгоритма, методика отладки алгоритмов, описание алгоритмов, иерархическая система, модернизируемость системы.

Для цитирования:

Васильченко Д. С. Принципы построения управляющих алгоритмов и их отладки для использования в системах управления движением судов / Д. С. Васильченко // Вестник Государственного университета морского и речного флота имени адмирала С. О. Макарова. — 2018. — Т. 10. — № 3. — С. 639–646. DOI: 10.21821/2309-5180-2018-10-3-639-646.

Введение (Introduction)

В современных управляющих компьютерных системах, в частности, применяемых на водном транспорте разработке накладывающих значительные ограничения на использование внешних ресурсов, вводится ограниченность параметров компьютерных систем [1] в соответствии с минимальными границами потребностей для решения требуемых задач управления. Задачи, реализуемые на управляющих компьютерных системах для управления объектами водного транспорта, сводятся к организации, контролю и обеспечению устойчивости вычислительного процесса в реальном масштабе времени. Полный список задач определяется назначением системы, однако особое внимание уделяется задачам обнаружения сигналов [2], прогнозирования изменения переменных, распределения ресурсов и адаптации процессов управления. Быстрое увеличение сложности и объёма программных приложений системы приводит к необходимости автоматизации и совершенствования технологии разработки программного обеспечения. В свою очередь, большая трудоёмкость отладки алгоритмов приводит к необходимости автоматизации этих процессов, поэтому вопрос особенностей и методов отладки систем управляющих алгоритмов требует особого внимания.

Методы и материалы (Methods and Materials)

Общая задача отладки алгоритмов состоит в установлении факта правильного функционирования и работоспособности программного обеспечения системы, а также в обнаружении ошибок, их диагностике и устранении. Методы решения этих задач и состав используемых средств в значительной степени зависят от типа управляющей системы [3] и объёма отлаживаемого программного обеспечения. Для реализации необходимо спроектировать комплекс специализированных методов, программного и аппаратного обеспечения, состоящий из модулей и объединяемый в подсистему отладки, представленную на рис. 1.

Архитектура подсистемы отладки определяется разработанной методикой отладки алгоритмов [4] и программных модулей системы управления, уровнем воздействия на отлаживаемое программное обеспечение, способом программной реализации подсистемы отладки, составом и характеристиками аппаратного обеспечения. Подсистема автоматизированной отладки позволит

сократить длительность и общие затраты на разработку системы управления. Однако создание сложных управляющих систем, включающих управляющие алгоритмы, практически невозможно без многоуровневого анализа и синтеза целей, задач и структур.

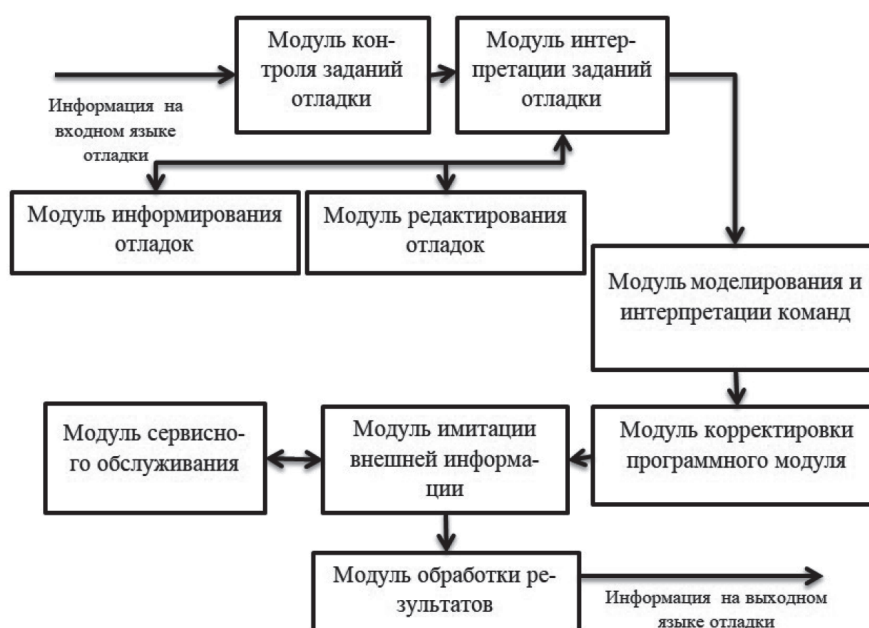


Рис. 1. Схема взаимодействия компонентов подсистемы отладки

Основой для описания алгоритмов являются: исходная информация, описываемая системой знаков; операторы, задаваемые на множестве исходной и результирующей информации; результирующая информация, заданная в конечном алфавите функция [5], связывающая операторы и правила их описания.

Любое сложное управляющее программное обеспечение системы можно разбить на иерархические уровни с определённой структурой. Иерархические системы характеризуются вертикальной декомпозицией с возможностью вмешательства программных модулей верхнего уровня для координирования функционирования программных модулей нижнего уровня, хотя в некоторых случаях функционирование модулей верхнего уровня зависит от информации, полученной от модулей нижнего уровня. Таким образом, сверху вниз идут координирующие и управляющие воздействия, а снизу вверх поступает информация о полученных решениях. На каждом уровне иерархии в сложных системах целесообразно иметь достаточную независимость элементов с минимальным взаимодействием в пределах уровня, когда управление сверху вниз заменяется автономно функционирующими элементами нижнего уровня. Это позволяет упростить общее координирование элементов системы и проводить его только по вертикали, а также сократить потоки информации, одновременно повышая надёжность, адаптивность и модернизируемость системы в целом, что приводит к снижению универсальности элементов, но повышает специфичность целей и решаемых задач. При этом необходимо учитывать, что многоуровневое иерархическое построение сложных систем влияет на продолжительность их разработки и отладки. Иерархическая структура требует определённых затрат ресурсов для элементов системы, но это оправдывается существенным упрощением анализа и синтеза всей системы.

Иерархическую структуру алгоритма [6] сложной системы управления также логично представить в виде взаимосвязанных подсистем нескольких уровней, каждый из которых построен по единому принципу. Алгоритм или команда более высокого уровня используют в качестве операторов алгоритмы или команды более низкого уровня. Следовательно, алгоритм или команда каждого иерархического уровня одновременно являются оператором для следующего уровня. Требуемые уровни представлены на рис. 2.

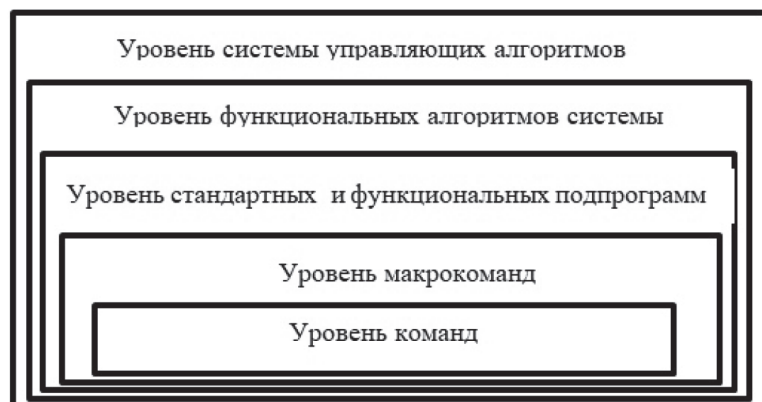


Рис. 2. Иерархия уровней алгоритмов и команд системы управления

Данная система уровней характеризуется последовательным усложнением функций и увеличением количества команд на каждом уровне иерархии. Переход на каждый более высокий уровень соответствует увеличению количества команд при любых целевых задачах управления, а увеличение уникальности совокупности команд, специализируясь, снижает возможность применения алгоритмов более высокого уровня в различных комбинациях. Данная структура позволяет существенно упростить анализ на каждом иерархическом уровне. Приведённое деление алгоритмов на иерархические уровни является условным, и выделенные границы не всегда чётко определены. Это касается и оценки сложности иерархических уровней по количеству команд. Алгоритмы в конкретных управляющих системах могут значительно отличаться при распределении по уровням от значения количества команд из-за сложности решения функциональных задач.

Иерархические уровни связаны не только передачей управления, но и обменом информацией, для каждого из которых определяются локальные переменные, используемые только на определённом уровне, что позволяет значительно ограничивать объём анализируемой информации по мере повышения операторов управляющего алгоритма. Переменные, используемые для взаимодействия алгоритмов и команд различных уровней, также образуют иерархическую структуру связей, которая является более сложной, чем структура связей по управлению [7].

Если рассматривать иерархию уровней снизу вверх (см. рис. 2), то, в первую очередь, необходимо отметить, что команда в управляющих компьютерных системах является наиболее используемым элементом управляющих алгоритмов. Алгоритм функционирования определяется набором типов операций. На первом этапе допустимыми процессами являются последовательные команды, образующие составные части и элементы реализации типовой операции системы. Выполнение каждой команды в программе рассматривается как реализация определённого алгоритма устройствами управляющей компьютерной системы. Именно уровень алгоритмов команд считается базовым и исходным.

На втором уровне операциями являются команды управляющей компьютерной системы, которые объединяются алгоритмами конкретных макрокоманд. Исходными данными являются величины, подлежащие анализу и преобразованию, объём которых в несколько раз больше, чем на предыдущем уровне. Вследствие этого увеличивается объём получаемых результатов, тем самым усложняя алгоритм функционирования. С повышением уровня часть результатов превращается в промежуточные данные, в связи с чем ограничивается скорость роста количества результирующих величин при повышении иерархического уровня. Эта возможность исключения из анализа промежуточных результатов при повышении иерархического уровня является одним из важнейших преимуществ многоуровневого иерархического построения управляющих алгоритмов. Количество типов макрокоманд в сложных алгоритмах того же порядка, что и количество типов команд.

Стандартные подпрограммы решают типовую вычислительную или логическую задачу, но большую по объёму, чем макрокоманда. Данный уровень характеризуется менее разнообразным

использованием. Из стандартных подпрограмм, используемых в качестве операторов, посредством команд и макрокоманд, образующих алгоритм функционирования, формируются функциональные подпрограммы, которые решают автономную частную целевую задачу алгоритма управления. Состав исходной информации в этом случае существенно больший, и увеличивается объём результирующих данных. Отсутствие необходимости раскрытия алгоритмов функционирования элементов, используемых в нижних иерархических уровнях, позволяет выполнять анализ подпрограмм в укрупнённом виде.

Уровень функциональных алгоритмов системы представлен алгоритмами, решающими целевую задачу, являющуюся частью целевой задачи управляющей системы. Функциональный алгоритм объединяет группу стандартных и функциональных подпрограмм, выступающих в нём в роли операторов. Количество исходной и результирующей информации больше, чем на предыдущих уровнях. Следует учитывать, что в некоторых случаях количество результирующей информации при переходе на более высокий уровень уменьшается, так как алгоритм осуществляет селекцию информации, используя промежуточные данные для получения частных результатов.

На уровне системы управляющих алгоритмов осуществляется построение элементов из функциональных алгоритмов и подпрограмм, которые используются как операторы, объединяемые операционной системой управляющей компьютерной системы. Результаты, получаемые системой управляющих алгоритмов, являются основными управляющими сигналами, предназначенными для решения главной целевой задачи управляющей системы в целом. Основными исходными данными на уровне системы управляющих алгоритмов являются сообщения управляемых объектов и внешних источников информации с текущей меткой времени. В каждый момент времени исходными данными является накопленная информация о результатах функционирования управляющей системы.

В рассмотренной иерархической структуре элементы различных уровней внутри одного уровня связаны между собой алгоритмически (взаимодействие по передачам управления) и информационно (взаимодействие через используемые переменные). Для сложных систем алгоритмов целесообразно иметь максимальную автономность элементов и стандартизированные связи между ними. Это обеспечивается тем, что связи по управлению для элементов основной части иерархических уровней строятся в простейшем виде, когда передача управления на включение подпрограммы или блока допускается только в одну точку. Аналогично выйти из блока можно только одним путём. Это позволяет построить более простые иерархические системы связи алгоритмов по управлению, что существенно упрощает их разработку и отладку. Однако иерархическое блочное построение больших алгоритмов может привести к значительному увеличению количества команд и производительности управляющей компьютерной системы, необходимой для реализации определённой совокупности задач управления. Эти потери обусловлены введением в алгоритмы элементов, необходимых для их унифицированного сопряжения [8], и затратами на организацию общего взаимодействия алгоритмов. Многократное использование отдельных подпрограмм и блоков снижает потери в памяти команд [9], однако они могут оставаться достаточно значительными. Требования к производительности управляющей компьютерной системы практически всегда возрастают при блочном иерархическом построении.

Потери вследствие блочно-иерархического построения алгоритмов обычно оказываются оправданными технологичностью проектирования и сокращением затрат на отладку системы алгоритмов [10]. В этом случае подпрограммы и другие элементы могут иметь автономный законченный вид и простые формализованные связи по информации и управлению. При развитии и модернизации системы алгоритмов управления такая структура позволяет корректировать или заменять отдельные алгоритмы или подпрограммы, не нарушая все остальные взаимодействующие алгоритмы. Кроме того, блочно-иерархическое построение существенно упрощает распараллеливание алгоритмов на различных уровнях для их реализации на многопроцессорных компьютерных системах. Отсутствие чёткой иерархической системы построения алгоритмов приводит к большому количеству внутренних связей, что существенно увеличивает длительность отладки

и её сложность. Более сложными представляются иерархические структуры управляющих алгоритмов по информационным связям. Это обусловлено тем, что в блоке или подпрограмме используются переменные, подготавливаемые различными блоками. Подпрограмма может рассчитывать величины для нескольких программ. Эти обстоятельства не позволяют построить простую последовательную схему подчинённости подпрограмм по информационным связям. Поэтому в структуре алгоритмов за основные принимаются связи по управлению, а информационные связи указываются как дополнительные.

Для создания управляющего алгоритма системы управления движением водного транспорта с возможностью отладки необходимо, в первую очередь, сформировать алгоритмы подпрограмм системы, в общем случае содержащие:

- обнаружение судов в зоне видимости;
- сопровождение судов;
- формирование решений конфликтов;
- индикацию движения водного транспорта;
- контроль конфликтных ситуаций;
- контроль движения на соответствие расписанию;
- подготовку и управление прибытием и отправлением судов;
- планирование движения судов;
- функциональный контроль системы;
- безопасность системы;
- тестирование системы;
- центральное управление системой;
- взаимодействие системы с внешними источниками;
- управление и контроль параметров системы.

Следующим этапом является выделение групп по управляющим алгоритмам, которые будут реализовываться с помощью следующих подпрограмм:

- 1) *алгоритмы управления вычислительным процессом*: центральное управление системой, взаимодействие системы с внешними источниками, управление и контроль параметров системы;
- 2) *общие функциональные алгоритмы обработки информации*: обнаружение судов в зоне видимости, сопровождение судов, формирование решений и устранение конфликтов;
- 3) *периодические функциональные алгоритмы*: индикация движения водного транспорта, контроль конфликтных ситуаций, контроль движения на соответствие расписанию, подготовка и управление прибытием и отправлением судов, планирование движения судов;
- 4) *алгоритмы контроля и обеспечения устойчивости решения задач*: функциональный контроль, безопасность и тестирование системы.

Далее осуществляется разработка функциональной схемы управляющего алгоритма системы управления движением судов, представленного на рис. 3, содержащая подсистему отладки.

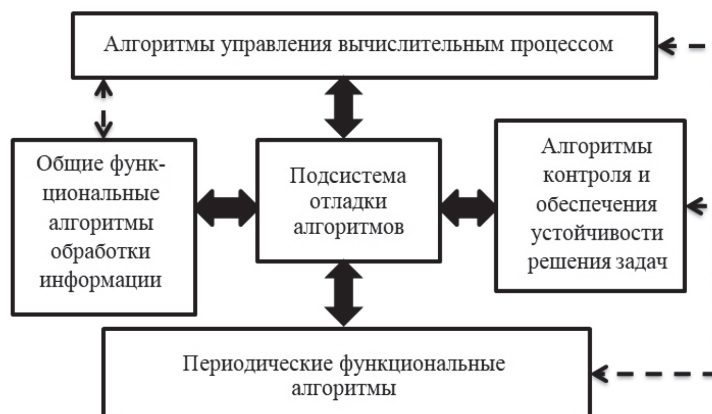


Рис. 3. Функциональная схема управляющего алгоритма системы управления движением судов

На данной схеме пунктирными стрелками обозначены связи групп алгоритмов системы, а стрелки с заливкой указывают на связь подсистемы отладки с алгоритмами. Подсистема отладки алгоритмов может быть как автоматической, так и выполняться по указанию пользователя системы.

Выводы (Summary)

1. Рассмотрены вопросы отладки алгоритмов и программных модулей системы управления движением судов. Представленный метод может быть применён при разработке как универсальных, так и информационно-поисковых систем управления. Однако его основным предназначением является использование при разработке систем управления объектами в реальном масштабе времени.

2. С повышением уровня алгоритмов и команд системы управления увеличивается её сложность и объём как используемой, так и результирующей информации. Причём количество используемых переменных при переходе на более высокий уровень увеличивается в несколько раз, однако темпы роста количества используемых переменных зависят от характера основной целевой задачи, решаемой управляемой системой. Информационная ёмкость различных переменных может существенно отличаться, поэтому иерархическая структура используемых и результирующих переменных является менее стройной, чем иерархическая структура алгоритмов.

3. Существенное влияние на рациональный состав разрабатываемой системы отладки оказывает количество и объём программных модулей, используемых для обеспечения требуемого функционирования. При разработке полной системы автоматизированной отладки сроки и трудоёмкость могут иметь очень большое значение, так как они способны внести корректировку в общий план работ по созданию общей системы управления. Одним из решений в данном случае будет являться использование системы сервисных программ для автоматизации отдельных технологических операций процесса разработки и отладки программных модулей системы.

СПИСОК ЛИТЕРАТУРЫ

1. *Моисеев А. Л.* Автоматизированная система контроля электрических параметров питания узлов компьютерных сетей / А. Л. Моисеев, В. В. Шаров, Р. Р. Моисеева, Ю. Н. Зацаринная // Вестник Казанского технологического университета. — 2013. — Т. 16. — № 11. — С. 240–241.
2. *Бурнаев Е. В.* Об одной задаче многоканального обнаружения сигналов / Е. В. Бурнаев, Г. К. Голубев // Проблемы передачи информации. — 2017. — Т. 53. — № 4. — С. 69–83.
3. *Легков К. Е.* К вопросу организации управляющей информационной сети автоматизированных систем управления инфокоммуникационными системами и сетями специального назначения / К. Е. Легков // Информация и космос. — 2015. — № 2. — С. 12–15.
4. *Федулов В. А.* Программное обеспечение для отладки алгоритмов работы современных станций обзора и целеуказания / В. А. Федулов // Молодежный научно-технический вестник. — 2017. — № 5. — С. 8.
5. *Иванников А. Д.* Формирование отладочного набора тестов для проверки функций цифровых систем управления объектами / А. Д. Иванников // Мехатроника, автоматизация, управление. — 2017. — Т. 18. — № 12. — С. 795–801. DOI: 10.17587/mau.18.795-801.
6. *Тельнова Е. В.* Хранение и обработка информации с помощью иерархических структур данных / Е. В. Тельнова, И. Н. Филоненко // Комплексные проблемы развития науки, образования и экономики региона: науч.-практ. журнал коломенского института (филиала) МГМУ (МАМИ). — 2016. — № 1 (8). — С. 136–146.
7. *Щекочихин О. В.* Объектно-процессная модель данных в управляющих информационных системах / О. В. Щекочихин // Научно-технический вестник информационных технологий, механики и оптики. — 2017. — Т. 17. — № 2. — С. 318–323. DOI: 10.17586/2226-1494-2017-17-2-318-323.
8. *Петров Р. В.* Варианты аппаратной реализации унифицированных принципов построения устройств сопряжения интерфейсов корабельных и авиационных систем / Р. В. Петров, А. Б. Токмаков // Морская радиоэлектроника. — 2016. — № 1 (55). — С. 40–43.

9. Бахтенко Е. А. Исследования особенностей организации машинной памяти в рамках разработки алгоритма верификации символов на стороне сервера / Е. А. Бахтенко, Е. О. Баланин // Таврический научный обозреватель. — 2016. — № 8–2 (13). — С. 6–9.

10. Липатников В. С. Отладка алгоритмов обработки радиолокационных сигналов в системе GNU RADIO / Липатников В. С., Чинёнов Д. А. // Цифровая обработка сигналов. — 2015. — № 4. — С. 63–68.

REFERENCES

1. Moiseev, A.L., V.V. Sharov, R.R. Moiseeva, and Yu.N. Zatsarinnaya. "Avtomatizirovannaya sistema kontrolya elektricheskikh parametrov pitaniya uzlov komp'yuternykh setei." *Vestnik Kazanskogo tekhnologicheskogo universiteta* 16.1 (2013): 240–241.

2. Burnaev, E.V., and G.K. Golubev. "Ob odnoi zadache mnogokanal'nogo obnaruzheniya signalov." *Problemy peredachi informatsii* 53.4 (2017): 69–83.

3. Legkov, K.E. "K voprosu organizatsii upravlyayushchei informatsionnoi seti avtomatizirovannykh sistem upravleniya infokommunikatsionnymi sistemami i setyami spetsial'nogo naznacheniya." *Informatsiya i kosmos* 2 (2015): 12–15.

4. Fedulov, V.A. "Programmnoe obespechenie dlya otladki algoritmov raboty sovremennykh stantsii obzora i tselekazaniya." *Molodezhnyi nauchno-tekhnicheskii vestnik* 5 (2017): 8.

5. Ivannikov, A.D. "Formirovanie otladochnogo nabora testov dlya proverki funktsii tsifrovyykh sistem upravleniya ob'ektami." *Mekhatronika, avtomatizatsiya, upravlenie* 18.12 (2017): 795–801. DOI: 10.17587/mau.18.795-801.

6. Telnova, E.V., and I.N. Filonenko. "Storage and processing of information using hierarchical data structures." *Kompleksnye problemy razvitiya nauki, obrazovaniya i ekonomiki regiona: nauchno-prakticheskii zhurnal kolomenskogo instituta (filiala) MGMU (MAMI)* 1(8) (2016): 136–146.

7. Schekochikhin, O.V. "Object-process data model in management information systems." *Scientific and Technical Journal of Information Technologies, Mechanics and Optics* 17.2 (2017): 318–323. DOI: 10.17586/2226-1494-2017-17-2-318-323.

8. Petrov, R.V., and A.B. Tokmakov. "Variants of the hardware implementation for the unified aufbau principles of the interaction interface means intended for interaction of the shipborne and aircraft systems." *Marine Radio-electronics* 1 (55) (2016): 40–43.

9. Bakhtenko, E.A., and E.O. Balanin. "Issledovaniya osobennostei organizatsii mashinnoi pamyati v ramkakh raz-rabotki algoritma verifikatsii simvolov na storone servera." *Tavricheskii nauchnyi obozrevatel* 8-2(13) (2016): 6–9.

10. Lipatnikov, V.S., and D.A. Chinenov. "Otladka algoritmov obrabotki radiolokatsionnykh signalov v sisteme GNU RADIO." *Tsifrovaya obrabotka signalov* 4 (2015): 63–68.

ИНФОРМАЦИЯ ОБ АВТОРЕ

Васильченко Дмитрий Сергеевич —
кандидат технических наук
ФГБОУ ВО «ГУМРФ имени адмирала
С. О. Макарова»
198035, Российская Федерация, Санкт-Петербург,
ул. Двинская, 5/7
e-mail: vasilchenkods@gumrf.ru

INFORMATION ABOUT THE AUTHOR

Vasilchenko, Dmitrii S. —
PhD
Admiral Makarov State University of Maritime
and Inland Shipping
5/7 Dvinskaya Str., St. Petersburg 198035,
Russian Federation
e-mail: vasilchenkods@gumrf.ru

Статья поступила в редакцию 16 мая 2018 г.
Received: May 16, 2018.