

16. He K. et al. Deep residual learning for image recognition, *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770-778.
17. Sandler M. et al. Mobilenetv2: Inverted residuals and linear bottlenecks, *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, ppP. 4510-4520.
18. Lin T., Maire M., Belongie S.J., Bourdev L.D., Girshick R.B., Hays J., Perona P., Ramanan D., Dollár P. and Zitnick C.L. Microsoft COCO: Common Objects in Context, 2014. DOI: 10.1007/978-3-319-10602-1_48.
19. Deng J., Dong W., Socher R., Li L., Kai Li and Li Fei-Fei. ImageNet: A large-scale hierarchical image database, *2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, 2009*, pp. 248-255.
20. Kingma D.P., Ba J. Adam: A method for stochastic optimization, *arXiv preprint arXiv:1412.6980*, 2014.

Статью рекомендовал к опубликованию к.т.н. П.К. Шульженко.

Тупиков Владимир Алексеевич – АО «Научно-производственное предприятие «Авиационная и морская электроника»; e-mail: tupikov@nppame.ru; 198097, г. Санкт-Петербург, ул. Маршала Говорова, 29, лит. «О», а/я 51; тел.: 88123274667; д.т.н.; профессор; зам. ген. директора; директор научно-производственного комплекса робототехнических систем специального назначения (НПК РТС СН).

Павлова Валерия Анатольевна – e-mail: pavlova@nppame.ru; к.т.н.; зам. директора НПК РТС СН по НИОКР.

Бондаренко Владимир Александрович – e-mail: bondarenko@nppame.ru; начальник центра средств интеллектуальной обработки изображений.

Холод Николай Григорьевич – e-mail: nikkholod@mail.ru; научный сотрудник.

Tupikov Vladimir Alekseevich – SPE "Research and Production Enterprise "Air and Marine Electronics"; e-mail: tupikov@nppame.ru; 198097, Saint Petersburg, street Marshal Govorov, 29, lit. "O", P.O.B 51; phone: 88123274667; dr. of eng. sc.; professor; Deputy Director General; Director of Research and production complex of special purpose robotic systems.

Pavlova Valeria Anatolyevna – e-mail: pavlova@nppame.ru; cand. of eng. sc.; Deputy Director of research at Research and production complex of special purpose robotic systems.

Bondarenko Vladimir Alexandrovich – e-mail: bondarenko@nppame.ru; chief of intellectual image processing center.

Holod Nicolai Grigorievich – e-mail: nikkholod@mail.ru; researcher.

УДК 004.896

DOI 10.18522/2311-3103-2020-1-199-218

В.П. Андреев, В.Л. Ким, С.Р. Эприков

АППАРАТНО-ПРОГРАММНЫЙ ФРЕЙМВОРК ДЛЯ РАЗРАБОТКИ МОДУЛЬНЫХ МОБИЛЬНЫХ РОБОТОВ С ИЕРАРХИЧЕСКОЙ АРХИТЕКТУРОЙ*

В настоящей работе рассматриваются основные проблемы, связанные с необходимостью интеграции разнородных робототехнических компонентов в единую систему при одновременном росте сложности навигационных алгоритмов мобильных роботов. Цель данной работы – представить иерархическую модульную архитектуру для реконфигурируемых мобильных роботов в качестве решения поставленных проблем. В этой архитектуре мобильный робот рассматривается как комбинация модулей, которые в свою очередь состоят из более простых блоков – субмодулей. Каждый субмодуль включает в себя мало-

* Работа выполнена при финансовой поддержке РФФИ (грант №19-07-00892а).

мощный микроконтроллер и отвечает только за базовые функции. Набор submodule образует модуль – транспортную платформу, ногу робота, манипулятор и т.д. Кроме того, одной из основных задач проекта является предоставление фреймворка, основанного на этой архитектуре, для быстрого прототипирования роботов из унифицированных модулей. В статье описываются изготовленные прототипы модулей, кратко рассматривается протокол межмодульного взаимодействия submodule, объединённых CAN-шиной. Представлены результаты экспериментов по тестированию протокола и приводится их анализ. Показана работоспособность предложенного решения, ограничения и краткий план дальнейших действий по реализации проекта.

Модульный робот; мобильный робот; распределённое управление; шинная топология; протокол коммуникации; CAN; компьютерные сети.

V.P. Andreev, V.L. Kim, S.R. Eprikov

HARDWARE-SOFTWARE FRAMEWORK FOR DEVELOPMENT OF MODULAR MOBILE ROBOTS WITH HIERARCHICAL ARCHITECTURE

In this paper, we consider the main problems associated with the need to integrate various robotic components into a single system while increasing the complexity of the navigation algorithms of mobile robots. The work aims to present a hierarchical modular architecture for reconfigurable mobile robots as a solution to the problems posed. In this architecture, a mobile robot is considered as a combination of modules, which in turn consist of simpler blocks - submodules. Each submodule includes a low-power microcontroller and is responsible only for the basic functions. A set of submodules forms a module - a transport platform, a robot leg, a manipulator, etc. Besides, one of the main objectives of the project is to provide a framework based on this architecture for rapid prototyping of robots from unified modules. The article describes the manufactured prototypes of the modules, briefly discusses the protocol of intermodule interaction of submodules connected by a CAN bus. The results of experiments on testing the protocol are presented and their analysis is given. The efficiency of the proposed solution limitations and a short plan of further actions for the implementation of the project are shown.

Modular robot; mobile robot; distributed control; bus topology; communication protocol; CAN; computer networks.

1. Введение. Одним из наиболее распространённых подходов к проектированию технических устройств и обеспечения их универсальности является разделение структуры технической системы на отдельные функциональные узлы – модули. Модульный подход широко известен в автомобильной индустрии, станкостроении и компьютерной технике. Например, в компьютерах почти все основные компоненты стандартизованы, и интерфейсы для их связи хорошо проработаны. Благодаря этому почти любой опытный пользователь может собрать персональный компьютер, используя совместимые компоненты-модули. В настоящее время модульные робототехнические системы активно развиваются, и они уже не ограничены лабораторными прототипами, но начинают появляться на рынке [1–2].

Модульный подход в мобильной робототехнике направлен на решение существенной проблемы – обеспечение интеграции и интероперабельности широкого спектра аппаратных и программных компонентов конкретной робототехнической системы (РТС). Современные специалисты-робототехники сталкиваются с общей проблемой «изобретения велосипеда» – значительная часть времени тратится на интеграцию аппаратных/программных интерфейсов вместо реализации новых алгоритмов.

Другая проблема заключается в том, что успешное выполнение роботами задач в реальных условиях требует от бортового компьютера выполнения большого числа математических вычислений с данными разных типов в реальном времени. Вычислительная нагрузка возрастает ещё больше при необходимости обработки видеопотоков и осуществления навигации с построением карты местности. Мо-

дальный подход к построению мобильных роботов может решить эту проблему, если вычисления, выполняемые на одном компьютере, будут распределены между вычислительными устройствами модулей.

Цель работы: представить аппаратно-программный фреймворк **ModRob** (ModRob framework) – иерархическую модульную архитектуру для мобильных роботов, позволяющую ускорить разработку РТС за счёт применения унифицированных компонентов (модулей). Предлагаемая архитектура является распределённой робототехнической системой с узлами, реализованными на встраиваемых вычислительных устройствах – микроконтроллерах или одноплатных компьютерах.

Статья организована следующим образом: в разделе 2 представлен краткий обзор аналогичных исследований, в разделе 3 подробно раскрывается предлагаемая модульная архитектура и основные виды разрабатываемых модулей, в разделе 4 кратко описан протокол межмодульного информационного взаимодействия и аппаратное обеспечение субмодулей, в разделе 5 описаны эксперименты и анализируются полученные результаты. В конце статьи приводятся выводы и заключение.

2. Обзор. Исследователи, занимающиеся модульными архитектурами роботов, стремятся достичь, в основном, двух целей:

1. Упростить работу других учёных и инженеров за счёт предоставления унифицированных робототехнических компонентов и/или универсальных программных интерфейсов. Как было отмечено ранее, сегодня имеется большое разнообразие несовместимых аппаратных и программных компонентов роботов, которые должны быть объединены в единую систему.

2. Разработка роботов для задач недетерминированного характера. Модульный подход позволяет создавать реконфигурируемые роботы, пригодные для выполнения миссий в обстановках с непредсказуемыми условиями.

Первая цель тесно пересекается с разработкой аппаратных и программных фреймворков, межплатформенного программного обеспечения (middleware) и операционных систем (ОС) для роботов (например, ROS). Фреймворки, межплатформенное программное обеспечение (ПО) и ОС преследуют одну цель – предоставить инструменты для быстрого прототипирования роботов. Проведённый аналитический обзор подобных систем показал, что большинство из них работает преимущественно с программным обеспечением роботов, не затрагивая аппаратную составляющую. Подробные обзоры существующих фреймворков уже опубликованы другими авторами, например [3]. Поэтому сконцентрируем наше внимание на исследованиях, включающих в себя не только программную, но и аппаратную часть модульных мобильных роботов.

В статье [4] представлена модульная мини робототехническая платформа AMiRo, в основе которой лежит идея распределения вычислительной нагрузки между различными модулями. Платформа содержит мощное аппаратное обеспечение и развитую открытую программную архитектуру. Стандартная конфигурация робота содержит три базовых модуля: *модуль движения* (управление движением, вычисление пройденного пути и т.п.), *модуль визуализации статуса и беспроводной связи*, а также *силовой модуль* (энергообеспечение). Модули имеют достаточно универсальный, но сложный электрический интерфейс: передача пяти различных уровней напряжения, поддержка 6 стандартов коммуникации (основной – CAN). Программная архитектура модулей достаточно нетривиальна, однако предоставляет широкие возможности для работы со сторонними приложениями. Тем не менее, добавление в систему модулей сторонних производителей может быть затруднено из-за многоуровневой архитектуры ПО и малых габаритов робота.

В работе [5] описывается набор модулей, преимущественно модульных приводов с последовательными упругими элементами, позволяющий ускорить разработку модульных *шагающих* роботов. Каждый модуль-привод управляет одним

бесщёточным двигателем постоянного тока по положению, скорости и моменту. Кроме приводных модулей есть также специализированные: сенсорные, модуль-шведское колесо, головной модуль. Все модули реализованы на микроконтроллерах серии ARM Cortex; межмодульное информационное взаимодействие осуществляется по сети с помощью семейства технологий Ethernet с использованием протоколов UDP/TCP. К достоинствам системы можно отнести распределённость вычислений, простоту реконфигурации, возможность подключения устройств сторонних производителей. Однако распределение вычислительных ресурсов выполняется лишь на нижнем (исполнительном) уровне модульной архитектуры системы управления робота, что не позволит решить проблему real time при усложнении целевой функции робота.

Интернациональный проект CLAWAR [6] был начат в 1997 году и координировался Университетом Портсмута. Проект CLAWAR разрабатывал концепцию функциональных модулей, требования и спецификацию к ним. Согласно концепции объектно-ориентированный подход использовался для методологии проектирования модульных компонентов. Проект описывает три основных типа модулей: человеко-машинный интерфейс, модули-актуаторы и сенсорные модули. По сравнению с другими подобными архитектурами Вирк Г.С. (Virk G.S.) представляет иерархическую модульную архитектуру: вся система разделяется на модули, которые в свою очередь разделяются на субмодули. Концепция CLAWAR достаточно перспективна, однако судя по информации из открытых источников, эта архитектура ещё не была реализована ни в одном из современных роботов.

В [7] представлена модульная архитектура для автономных мобильных роботов, основанная на так называемой «непрерывной» модульности аппаратного и программного обеспечения. Аппаратное обеспечение состоит из четырёх уровней: уровень движения, сенсорный уровень, актуаторный уровень и управляющий уровень. Межмодульная коммуникация в роботе основана на CAN-шине. Программная архитектура реализована на межплатформенном ПО MIRO. Эффективность роботов была показана на соревновании RoboCup 2003, для которого серия роботов была изготовлена меньше чем за год. Несмотря на наличие распределённых вычислений, авторы не раскрывают детали коммуникационного процесса между модулями, что не позволяет судить о наличии свойства реконфигурируемости с использованием модулей сторонних производителей.

Модульные реконфигурируемые антропоморфные роботы представлены в работах [8–10]. В [8] распределённые вычисления выполняются на функциональных модулях: модуль-голова, главный модуль, левая/правая рука, колёсный модуль. Авторы разработали собственный коммуникационный интерфейс реального времени Responsive Link для межмодульного взаимодействия. Кроме этого коммуникационный интерфейс также включает в себя два USB порта и порта IEEE 1394. Используя предложенную архитектуру, авторы изготовили реконфигурируемый антропоморфный робот. Основным недостатком системы заключается в необходимости использования специального встраиваемого процессора Responsive Processor, способного поддерживать интерфейс Responsive Link, что осложняет интеграцию компонентов сторонних производителей. В работе [9] предложен аналогичный подход к проектированию гуманоидного робота. Согласно этому подходу все модули являются полнофункциональными устройствами, причём модули способны взаимодействовать между собой, даже если на их вычислителях установлены различные операционные системы. Свойство достигается за счёт программной архитектуры и специальной архитектуры программного соединения, определяю-

шей 5 стандартов информационных моделей. Межмодульная коммуникация основана на TCP/IP и RS485. Хилд М. (Hild M.) и Спрангер М. (Spranger M.) также разработали модульный антропоморфный робот Муон с распределённой управляющей архитектурой DISTAL [10]. Основные особенности робота: автономная работа модулей и возможность реконфигурации даже при включённом роботе. Для межмодульного информационного взаимодействия авторы разработали две коммуникационные шины: Spinal Cord («спинной мозг») и Extended Spinal Cord (расширенный спинной мозг»). К недостаткам системы можно отнести отсутствие программного интерфейса между модулями, а также наличие распределённых вычислений лишь на уровне управления приводами.

Концепция модульного сервисного робота DRP I представлена в работе [11]. DRP I состоит из полнофункциональных модулей на одноплатных компьютерах и дополнительных электронных блоков для управления и тестирования. ПО каждого модуля состоит из уровня ОС, уровня виртуальной машины, прикладного программного интерфейса (API) и прикладного приложения. Межплатформенное ПО робота основано на IEEE 1394. Реализовано «горячее» подключение модулей робота с автоматической реконфигурацией навигационных алгоритмов. Несмотря на все достоинства представленной архитектуры, отсутствуют современные робототехнические решения, основанные на ней.

Группа исследователей в работе [12] представила парадигму построения распределённых архитектур систем управления мобильных роботов – ASEBA. ASEBA представляет собой полностью централизованную систему, состоящую только из микроконтроллеров, управляющих датчиками и приводами. Все модули (микроконтроллеры) объединены CAN-шиной. В ASEBA реализовано управление по событиям за счёт виртуальных машин, запускаемых на микроконтроллерах. С нашей точки зрения данная архитектура не обеспечит работу РТС в реальном масштабе времени в случае выполнения даже не очень сложных работ.

В статье [13] представлена система R2P, которая имеет сходство с нашим проектом. R2P – открытый аппаратно-программный фреймворк, позволяющий комбинировать различные модули в режиме «plug-and-play». Модули R2P реализованы на микроконтроллерах ARM Cortex-M3, взаимодействующих по шине CAN с использованием авторского протокола RTCAN. Недостатком RTCAN является фрагментация всех сообщений, что может приводить к коллизиям CAN-фреймов. Также из приведённого в статье материала неясно, по какому принципу тот или иной программный узел R2P должен быть развёрнут на аппаратном модуле, что не приводит к пониманию требований к вычислительным ресурсам модулей.

Имеющий сходства с R2P фреймворк RoboCAN [14] также представляет собой модульную архитектуру на CAN-шине. RoboCAN в свою очередь является частью фреймворка RIDE. Все модули основаны на микроконтроллерах и поддерживают режим «plug-and-play». Стоит отметить, что в RoboCAN используется собственный протокол, который незначительно лучше общеизвестного CANOpen: полоса пропускания коммуникационного канала в RoboCAN лишь на 25% меньше чем в CANOpen (при неработающих приводах). Также нет ясности относительно устойчивости данного протокола к образованию коллизий и относительно достаточности его пропускной способности для обеспечения межмодульного информационного взаимодействия при различных режимах работы РТС.

В статье [15] представлен набор для прототипирования модульных мобильных роботов преимущественно в образовательных целях. Набор включает различные блоки-модули: управляющий модуль, блок питания, приводные модули, мо-

дули передачи данных, редукторы и т.д. Модули соединяются по шине RS485 либо напрямую к управляющему модулю. Модули набора имеют встроенные электронные/электромеханические устройства (электронная плата модуля не отвязана от исполнительного устройства). Такой подход позволяет легко собирать мобильные роботы, однако ограничивает область применения набора: например, если пользователю понадобятся более мощные двигатели или иные датчики.

Амбициозный проект H-ROS представлен в статье [16]. Цель проекта – разработать универсальный аппаратно-программный фреймворк для обеспечения интеграции и интероперабельности различных компонентов роботов. H-ROS основан на ROS, но использует функциональность новейшей ОС для роботов – ROS 2. Физический уровень межмодульной коммуникации базируется на EtherCAT. Над физическим уровнем располагается ядро Linux с патчем (patch) реального времени PREEMPT_RT. К недостаткам системы можно отнести необходимость использования достаточно мощного в вычислительном плане одноплатного компьютера в *каждом* модуле сети (для развёртывания Linux и ROS 2), что избыточно для модулей, выполняющих низкоуровневое управление исполнительными устройствами. Кроме того, для межмодульного информационного взаимодействия используется достаточно дорогая технология EtherCAT. В настоящий момент, проект H-ROS разработчиками остановлен.

В работе [17] представлена концепция модульной архитектуры роботов на базе так называемого Модуля Оператора-Контроллера (Operator-Controller Module – OCM). Новизна концепции заключается в модульной иерархической структуре робота и организации распределённых вычислений на локальной части (сам робот) и удалённой части (облачные вычисления). Удалённая часть значительно увеличивает вычислительную мощность системы для комплексных алгоритмов без повышения энергопотребления робота, однако существенно снижает автономность мобильной части и надёжность ПТС. Авторы разработали прототип мобильного робота с распределённой системой управления, включающей в себя два одноплатных компьютера Raspberry Pi, Arduino Mega, платформу STM32 Discovery и ПЛИС ZynqBerry TE0726-03M. Все устройства кроме ПЛИС объединены CAN-шиной. Отметим, что эта архитектура никак не унифицирует компоненты (модули) робота, что может усложнять реконфигурацию.

Краткий анализ представленных работ показал, что далеко не все архитектуры и фреймворки развиты настолько, чтобы обеспечить надёжным инструментом других специалистов для быстрого прототипирования робототехнических решений. При этом, как правило, реконфигурируемость обеспечивается в пределах предлагаемого комплекта модулей, что не позволяет в полной мере реализовать возможность применения модулей сторонних производителей. Кроме того, большая часть работ рассматривает распределённые вычисления только на низком уровне (приводном), в то время как реализация распределённых вычислений наиболее сложных алгоритмов управления и комплексирования данных практически не рассматривается. Также отметим, что некоторые из представленных систем требуют использования мощных одноплатных компьютеров, что отчасти ослабляет положительный эффект модульного подхода.

3. Иерархическая модульная архитектура мобильных роботов. В рамках данного проекта предлагается использование пирамидальной модульной архитектуры для мобильных роботов различного назначения [18]. На рис. 1 показана схема такой обобщённой иерархической структуры информационно-измерительной и управляющей системы (ИИУС).



Рис. 1. Обобщённая иерархическая структура ИИУС модульного мобильного робота

На вершине архитектуры располагается модуль интеллектуального управления, отвечающий за планирование и распределение задач между модулями второго уровня (общесистемное управление). В этой архитектуре модули на третьем и втором уровне иерархии отвечают за одну укрупнённую функцию целого робота согласно принципу полной функциональности [19]: *каждый модуль робота должен быть способен любым удобным ему способом выполнять свою целевую функцию, используя только собственные средства для выполнения команд от внешней системы управления*. В общем случае структура системы управления типового мобильного робота может быть разделена на следующие основные функциональные модули [20]:

1. **Модуль интеллектуального управления** – планирование и распределение задач между модулями второго уровня (общесистемное управление).
2. **Транспортный модуль** – перемещение робота в пространстве (транспортная функция).
3. **Силовой модуль** – обеспечивает электропитание компонентов робота (энергетическая функция).
4. **Дистанционный сенсорный модуль** – поиск препятствий и объектов манипулирования (информационная функция).
5. **Модуль воздействия на среду** – технологическая функция.
6. **Модуль организации беспроводной связи** – коммуникация с внешним супервизором.

Разбиение структуры системы управления мобильного робота именно на такие блоки обусловлено наличием всех перечисленных выше функций практически в любом мобильном роботе с элементами интеллектуального или адаптивного управления [21-23]. Например, в [21] рассматривается обобщённая структурная схема системы управления «очувствлённым» роботом, которая включает в себя уровень общего функционирования (*общесистемное управление*), сенсорные устройства (*информационная функция*), технологические операции (*технологическая*

функция), управление приводами (*транспортная и технологическая функция*), а также пульт управления оператора (*коммуникационная функция*). В модульном роботе блоки, отвечающие за аналогичные функции, ещё обязательно должны взаимодействовать между собой напрямую или опосредованно для согласованного функционирования системы.

Некоторые модули второго уровня должны выполнять большой объём работы. Например, транспортный модуль должен отвечать за планирование траектории движения, объезд препятствий, управление приводами, расчёт пройденного пути. Транспортный модуль выполняет все эти задачи в кооперации с сенсорным модулем, который тоже загружен достаточно трудоёмкими задачами: фильтрация и комплексирование показаний датчиков, построение карты местности и т.д.

Согласно нашей концепции построения архитектуры системы управления ModRob каждый полнофункциональный модуль также должен быть построен как модульная архитектура с узлами-субмодулями. Такая архитектура позволит распределить вычислительную нагрузку различной функциональной значимости между субмодулями, особенно в тех случаях, когда число приводов и датчиков может увеличиваться или их тип может изменяться. Дополнительно появляется возможность комбинировать модули как на первом, так и на втором уровнях иерархии соответственно.

Рассмотрим распределённую архитектуру системы управления полнофункционального **транспортного модуля** (рис. 2).

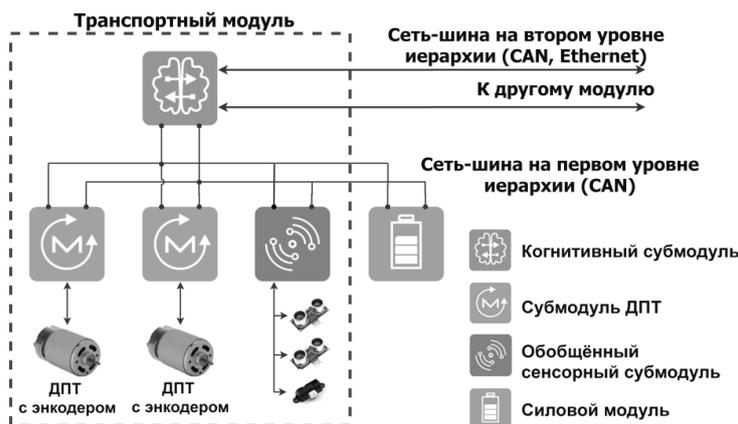


Рис. 2. Распределённая архитектура системы управления транспортным модулем

В этой иерархической структуре взаимодействие транспортного модуля с остальными полнофункциональными узлами робота имеет место в сети на втором уровне иерархии (например, CAN, Ethernet, USB). Предполагается, что это взаимодействие может выполняться в режиме «мягкого» реального времени.

Структура транспортного модуля основана на трёх типах субмодулей:

1. **Когнитивный субмодуль** – отвечает за управление и настройку остальных субмодулей.
2. **Субмодуль-привод** – управляет работой подключаемого к нему электродвигателя (на рис. 2 представлен частный случай – субмодуль для управления двигателем постоянного тока).
3. **Обобщённый сенсорный субмодуль** – позволяет подключать различные датчики расстояний, гироскопы, акселерометры и т.д.

Наличие обобщённого сенсорного субмодуля в транспортном модуле является опциональным; он может быть использован в тех ситуациях, когда сенсорный модуль (на верхнем уровне иерархии) отсутствует в мобильном роботе и существует высокий риск столкновения с препятствиями.

На рис. 2 также представлен **силовой модуль**, обеспечивающий электропитание всех подключённых устройств. Силовой модуль не является частью транспортного модуля и рассматривается как специфичный узел представленной системы, поскольку информационное взаимодействие он осуществляет с *модулями* (второй уровень иерархии), но электропитанием обеспечивает как модули, так и *субмодули*.

Информационное взаимодействие между субмодулями должно осуществляться как минимум в «мягком» реальном времени (например, CAN, I2C или RS485).

Рассмотрим подробнее устройство субмодулей.

Когнитивный субмодуль (Cognition) – вычислительное устройство, основные функции которого программируются пользователем. Например, когнитивный субмодуль может отвечать за установку линейной и угловой скоростей транспортного модуля согласно алгоритму, расчёт позиции робота в пространстве и т.д. Кроме этого когнитивный субмодуль исполняет роль «моста» между компьютером высокого уровня и другими субмодулями (рис. 3). Это необходимо для настройки субмодулей: установки коэффициентов ПИД-регуляторов, параметров фильтрации и интерполяции и т.д. Подключение этого субмодуля к компьютеру может осуществляться по проводной (USB) и беспроводной связи (Wi-Fi).

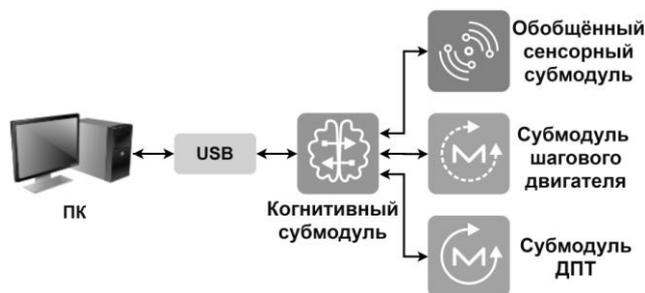


Рис. 3. Настройка субмодулей с помощью компьютера

Субмодуль-привод (Actuator) – электронный контроллер, управляющий различными электродвигателями: коллекторными, бесколлекторными двигателями постоянного тока (ДПТ), шаговыми двигателями, сервомоторами. Основные функции субмодуля: управление валом по скорости, положению и/или моменту. В зависимости от типа регулируемого электродвигателя субмодули-приводы могут быть четырёх основных видов: контроллер коллекторного ДПТ, контроллер бесколлекторного ДПТ, контроллер шагового двигателя, контроллер сервомоторов.

Обобщённый сенсорный субмодуль (General Sensor Submodule) – электронное устройство, предназначенное для подключения различных датчиков, преимущественно дальномеров, энкодеров, концевых выключателей, а также инерционных датчиков (гироскопов, акселерометров). Основные функции субмодуля: сбор и обработка сенсорных данных, комплексирование. Важной особенностью субмодуля является возможность подключения к нему различных моделей датчиков одного и того же типа, например, ультразвуковых дальномеров. Предполагается, что субмодуль должен быть совместим с наиболее распространёнными типовыми сенсорами (рис. 4).



Рис. 4. Подключение различных типовых датчиков к обобщённому сенсорному submodule

4. Лабораторные прототипы submodule и протокол взаимодействия.

В ходе работы над проектом были изготовлены прототипы *когнитивного submodule*, submodule двигателя постоянного тока (ДПТ) и силового модуля. Также были разработаны и изготовлены так называемые **базовые submodule** для проведения тестов и отладки протокола коммуникации.

Все разработанные прототипы submodule имеют следующие основные компоненты: в качестве основного вычислителя – микроконтроллер STM32 ARM-Cortex M3 с частотой 72 МГц, 20 КБ ОЗУ и 64 КБ флеш-памяти, драйвер для CAN-шины TJA1050 и понижающий DC-DC преобразователь напряжения.

Когнитивный submodule: прототип (помимо базовых компонентов) включает в себя контроллер Wemos D1 mini для организации беспроводной связи между модульным роботом (транспортным модулем) и компьютером пользователя. Для отладки когнитивного и других submodule используется UART интерфейс.

Submodule ДПТ: текущий прототип способен управлять двигателями с питающим напряжением до 35 В и номинальным током до 4 А. Замкнутое управление с обратной связью по положению вала и/или его скоростью реализуется, если энкодер подключен к submodule. Submodule ДПТ согласно протоколу межмодульного информационного взаимодействия публикует на шину информацию о скорости и положении вала с заданной частотой (частота публикации определяется когнитивным submodule или компьютером пользователя). На рис. 5 показан первый вариант прототипа submodule ДПТ.

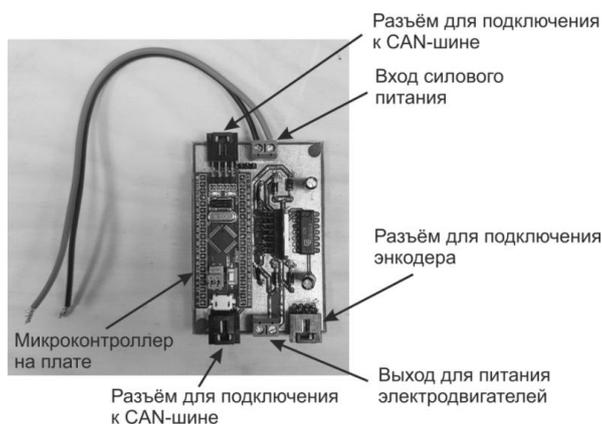


Рис. 5. Первый вариант прототипа submodule ДПТ

Силовой модуль: текущий прототип модуля включает в себя литий-ионные аккумуляторы, два микроконтроллера ATtiny85, DC-DC преобразователь напряжения и вспомогательные электронные компоненты. Выходное напряжение модуля лежит в диапазоне 10–16.8 В, величина тока ограничена 5 А в продолжительном режиме, и 10 А – в кратковременном. В текущих экспериментах используются маломощные моторы, поэтому данные характеристики вполне удовлетворяют поставленным требованиям.

Для *тестирования протокола межмодульной коммуникации и отладки алгоритмов управления submodule* было разработано несколько базовых submodule.

Базовый submodule (BASE_0) представляет собой электронное устройство, включающее в себя только основные компоненты, присущие всем submodule: микроконтроллер, драйвер CAN-шины и силовой преобразователь. Также на плате устройства имеются разъёмы для подключения к шине и вспомогательные электрические контакты, дублирующие выводы платы с микроконтроллером. На рис. 6 приведена фотография четырёх изготовленных базовых submodule, подключенных по шине.

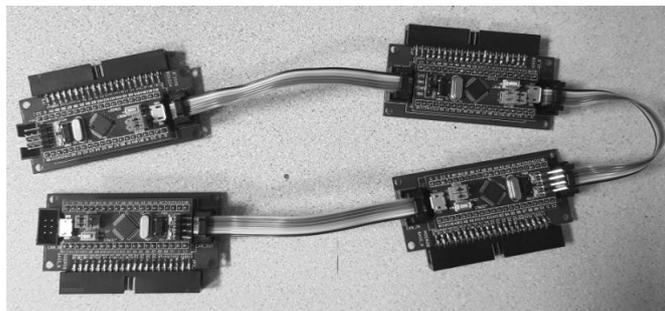


Рис. 6. Базовые submodule, подключенные к шине CAN с помощью шлейфа

Базовый submodule является *универсальным устройством*, способным выполнять функции остальных submodule при подключении необходимой периферии и загрузки соответствующей управляющей программы.

Все submodule подключаются к CAN-шине. Выбор этого стандарта обусловлен следующими его достоинствами: возможность обмена сообщениями в реальном времени, простота реализации, высокая помехоустойчивость, обнаружение коллизий и контроль ошибок. Кроме того, CAN-шина позволяет любому модулю становиться мастером благодаря арбитражу доступа к сети. К недостаткам можно отнести малый объём полезной нагрузки (всего 64 бита) и небольшую пропускную способность шины (до 1 Мбит/с). Тем не менее, именно при иерархической организации системы управления модульного мобильного робота при соблюдении принципа полной функциональности модулей объём межмодульного информационного взаимодействия оказывается небольшим, что позволяет использовать такой хорошо зарекомендовавший себя стандарт.

Рассмотрим кратко *протокол межмодульного взаимодействия* (на первом – нижнем уровне иерархии).

На каждом аппаратном submodule развёрнут программный узел (node). Протокол основан на популярном принципе «подписчик-издатель» (publisher-subscriber). Каждый узел имеет свой идентификатор (modID) и набор перемен-

ных со своими идентификаторами (varID). Программный узел способен принимать и публиковать сообщения. Каждое сообщение занимает только *один* расширенный CAN-фрейм (128 бит). Сообщение содержит следующую основную информацию: адрес модуля, бит интерпретации адреса, значение переменной и адрес типа модуля.

Сообщения приходят ко всем узлам шины. Входящее сообщение интерпретируется либо как «установка» (set), либо как «обновление» (update). В зависимости от режима **сообщение-установка** может:

- ◆ *записать значение* в переменную приёмника (по ID переменной);
- ◆ *установить частоту публикации* на шину какой-либо переменной приёмника (по ID);
- ◆ *подписать переменную* приёмника на какую-либо переменную передатчика (по двум ID).

Сообщение-обновление действует следующим образом: переменная приёмника, которая ранее была подписана на переменную передатчика (сообщением-установкой), получает новое значение (*обновляется*).

Когнитивный submodule, регулирующий работу остальных submodule, подключается к компьютеру верхнего уровня по отдельному каналу коммуникации. В настоящий момент таким каналом является USART. С компьютера на когнитивный submodule («вниз») команды отправляются по последовательному порту (USB-USART). Такими командами могут быть, например, задание целевой скорости вращения вала двигателя submodule ДПП.

Команды от компьютера аналогичны сообщениям-установкам: имеют схожие информационные поля. Все команды записаны в формате JSON. Такой выбор обусловлен широкой распространённостью формата и лёгкостью его восприятия человеком.

Используются всего три вида сообщений:

1. {"op": "setValue", "modID": Int16, "varID": Int8, "value": Float} – установка значения переменной с идентификатором varID модуля с идентификатором modID;
2. {"op": "setPublication", "modID": Int16, "varID": Int8, "frequency": Int} – установка частоты (Гц) публикации переменной с идентификатором varID, принадлежащей модулю с идентификатором modID (0 – нет публикации);
3. {"op": "setSubscription", "modID": Int16, "varID": Int8, "subModID": Int16, "subVarID": Int8} – установка подписки: переменная с идентификатором varID модуля с идентификатором modID подписывается на переменную с идентификатором subVarID модуля с идентификатором subModID, т.е. при изменении переменной subVarID будет изменяться переменная с идентификатором varID, если до этого была установлена *ненулевая* частота публикации переменной с идентификатором subVarID.

Отправка сообщений, публикуемых на CAN-шине «наверх» (к компьютеру верхнего уровня), осуществляется за счёт агрегирования данных на когнитивном submodule. Агрегированные данные также представляются в формате JSON.

5. Экспериментальная часть. Было выполнено тестирование работы submodule с использованием разработанного протокола.

Цель эксперимента: исследовать возможность использования разработанного протокола для управления submodule в реальном времени. Схема эксперимента представлена на рис. 7.



Рис. 7. Схема экспериментальной установки: а – контур скорости субмодуля ДПТ замкнут на самом субмодуле ДПТ; б – контур скорости субмодуля ДПТ замыкается через промежуточный субмодуль

Состав оборудования для проведения эксперимента представлен в табл. 1.

Таблица 1

Состав оборудования для эксперимента

Оборудование	Количество	Основные характеристики
Ноутбук (компьютер верхнего уровня)	1	Операционная система Manjaro Linux 19.0.0
Базовый субмодуль	3	См. раздел 4
Силовой преобразователь L298N	1	Напряжение питания: 5-46 В, максимальный продолжительный ток: 2 А
Интегральная микросхема K155ЛЕ1	1	Четыре логических элемента 2ИЛИ-НЕ
Оптический инкрементальный энкодер	1	Минимальное расстояние между метками: 0.229 мм, напряжение питания: 5 В
Кодовый диск	1	Разрешающая способность: 30 имп/об.
Электродвигатель постоянного тока Raboesch Blue RM-410	1	Номинальное напряжение питания: 7.2 В, номинальная скорость: 7500 об/мин, номинальный ток обмотки якоря 1.3 А, номинальный крутящий момент: 0.022 Нм

Замечание: в составе оборудования нет субмодуля ДПТ, тем не менее, на схеме на рис. 7 он присутствует. Это объясняется тем, что субмодуль ДПТ в этом эксперименте реализован на основе базового субмодуля, а также силового преобразователя L298N и интегральной микросхемы K155ЛЕ1.

Постановка эксперимента. Осуществляется управление ДПТ по скорости вращения вала. Эксперимент разделён на два опыта. В первом опыте (рис. 7,а) электродвигатель с энкодером подключены к субмодулю ДПТ (идентификатор ID 1), т.е. обратная связь по скорости замыкается на этом субмодуле. Во втором опыте энкодер подключается к другому субмодулю с ID 2 (рис. 7,б), т.е. обратная связь по скорости замыкается с помощью этого субмодуля. И в первом и во втором опыте задание целевой скорости вращения вала, а также установка частот публикации и подписок осуществляется с помощью компьютера, подключенного к первому базовому субмодулю (ID 0) по USART.

Первый опыт. Субмодуль ДПТ измеряет скорость вращения вала двигателя и записывает её величину в переменную. Командой "setPublication", отправляемой компьютером, устанавливается частота публикации переменной на шину – 100 Гц. Привод субмодуля ДПТ должен обрабатывать ступенчатые задающие воздействия, подаваемые на его вход: 100, 500, 800, 250, -250, -800, -500, -100 рад/с. Желаемая скорость задаётся командой "setValue". В приводе реализован только контур скорости с ПИД-регулятором. Отметим, что ПИД-регулятор не настроен на оптимальный режим работы. В табл. 2 представлены исходные параметры лабораторной установки для первого опыта.

Таблица 2

Исходные параметры лабораторной установки для первого опыта

Параметр	Значение
Пропускная способность CAN-шины, Мбит/с	1
Скорость работы последовательного порта (USART), бод/с	115200
Частота работы контура скорости в субмодуле ДПТ, Гц	100
Частота публикации переменной скорости на шину, Гц	100
Коэффициент усиления пропорциональной составляющей	3.5
Коэффициент усиления интегральной составляющей	9
Коэффициент усиления дифференциальной составляющей	0.2

Замечание: измеряемая скорость усредняется по 10 точкам.

На рис. 8 непрерывной линией показан график изменения скорости вращения вала электродвигателя. Штриховой линией показан график изменения задающего воздействия (желаемой скорости).

Второй опыт. Как и в первом опыте, привод субмодуля ДПТ должен обрабатывать ступенчатые задающие воздействия, подаваемые на его вход. Измерение скорости производится в субмодуле с ID 2 (к которому подключен энкодер); публикация выполняется с частотой 100 Гц. Для замыкания обратной связи субмодуль ДПТ подписывается на переменную скорости субмодуля с энкодером с помощью команды "setSubscription".

Исходные параметры лабораторной установки для второго опыта такие же, как и для первого.

На рис. 8 штрихпунктирной линией показан график изменения скорости вращения вала электродвигателя, полученный в ходе второго опыта.

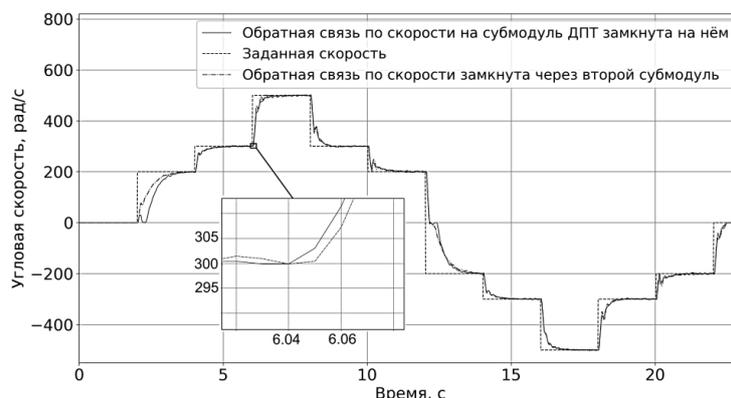


Рис. 8. Сравнение скоростей в первом и втором опыте

Из графиков на рис. 8 видно, что качество переходных процессов изменения скорости во втором опыте практически не изменилось по сравнению с первым опытом, однако к задержке реакции привода на задающее воздействие добавилось запаздывание по сравнению с первым опытом. Запаздывание вызвано наличием субмодуля в цепи обратной связи. Основные причины запаздывания: измерение скорости с периодом 10 мс, публикация с периодом 10 мс, а также обработка приёмником входящих сообщений, которая происходит с периодом 10 мс. Время, затрачиваемое на отправку сообщения по шине (порядка 14–16 мкс), вносит незначительный вклад в общую длительность запаздывания. Таким образом **максимальное запаздывание** должно быть порядка 40–50 мс.

Анализ графиков показал, что **реальное запаздывание** варьируется в диапазоне от 4 мс до 50 мс, что соответствует расчёту. В начальный момент времени (при задающем воздействии 200 рад/с), задержка реакции во втором опыте оказалась даже меньше, чем в первом (на 0.25 с).

Также появились незначительные колебания (но в рамках установившейся ошибки в 5 %) – по 2–3 полных колебания за время регулирования на скоростях более 200 рад/с.

Опыты были повторены для синусоидального задающего воздействия с частотой 0.5 Гц:

$$\omega_z = 500 \cdot \sin(\pi t),$$

ω_z – заданная скорость, рад/с.

Все остальные параметры эксперимента остались прежними, кроме параметров ПИД-регулятора, которые стали равными $K_P = 6$, $K_I = 20$ и $K_D = 0.3$. На рис. 9 показаны графики изменения скорости вращения вала двигателя для первого и второго опыта соответственно при синусоидальном воздействии.

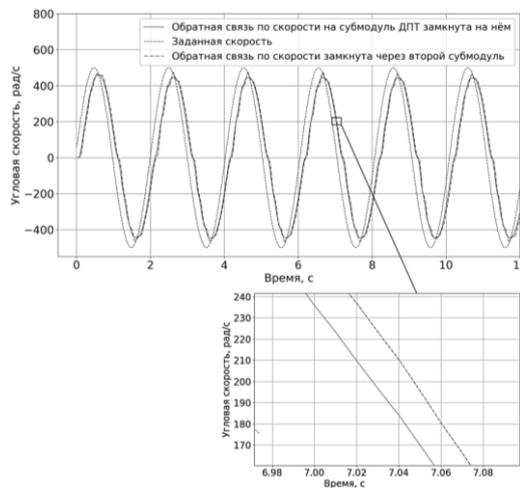


Рис. 9. Сравнение скоростей при подаче синусоидального воздействия

Как и в предыдущем случае, скорости в первом и во втором опытах отличаются незначительно, тем не менее, также присутствует задержка реакции привода. Задержка варьируется в диапазоне от 13 до 25 мс (в начальный момент времени). Средняя величина задержки – 20 мс.

Вывод по эксперименту. Анализ результатов показал, что использование разработанного протокола межмодульной коммуникации позволяет осуществлять управление субмодулями в реальном времени. Строго говоря, поставленный экс-

перимент является *искусственным*, поскольку согласно представленной концепции submodule ДПТ самостоятельно должен рассчитывать скорость электродвигателя. Тем не менее, введение дополнительного submodule в цепь обратной связи по скорости не нарушило функционирование системы, хотя и ухудшило его в допустимых пределах. Привод сохраняет устойчивость, однако возросшая задержка реакции на задающее воздействие может негативно сказаться на работе привода. Это особенно важно в высокоскоростных приводах *промышленных* роботов, но поскольку разрабатываемый нами фреймворк нацелен в первую очередь на *мобильные* роботы, то этот недостаток не является критичным. Скорость большинства мобильных роботов, работающих в помещениях с людьми, редко превышает 1–2 м/с [24–26]. В данном случае задержка реакции робота (транспортного модуля) на задающее воздействие может приводить к позиционной ошибке порядка 0.8–10 см, что не должно являться большой проблемой при наличии дистанционных датчиков, публикующих данные на шину с той же частотой, что и submodule с энкодером (100 Гц).

Добиться уменьшения задержки реакции привода в этом случае можно за счёт увеличения частоты публикации переменной скорости на шину. В то же время следует во столько же раз увеличить частоту измерения скорости на submodule с ID 2 (см. рис. 7,б). Однако улучшения качества работы можно достичь, только если кодовый диск энкодера обладает большим разрешением. Для использованного в данном эксперименте кодового диска с разрешением 30 имп/об увеличение частоты публикации до 200 Гц только ухудшило качество работы привода.

Стоит также отметить, что увеличивать частоту публикации переменных того или иного submodule можно лишь до определённого предела, поскольку CAN-шина будет загружена сообщениями других submodule, которых в одном транспортном модуле может быть 5–10 шт. Анализ работы протокола с большим числом активно публикующих на шину submodule является частью дальнейших исследований.

Выводы. В настоящей статье предложен фреймворк Modrob для разработки мобильных роботов с модульной иерархической архитектурой. К основным отличиям данного фреймворка от приведённых в обзоре можно отнести:

- ◆ иерархическая топология, обеспечивающая распределение задач между модулями как по «вертикали», так и по «горизонтали»;
- ◆ использование полнофункциональных модулей на всех уровнях иерархии: начиная от низкоуровневого управления приводами и заканчивая высокоуровневым управлением мобильными платформами.

Показано, что второй уровень иерархии может быть представлен шестью основными полнофункциональными модулями, каждый из которых должен включать в себя несколько типовых submodule, что должно обеспечить *лёгкость интеграции устройств сторонних производителей*.

Разработана структура транспортного модуля, состоящего из трёх основных типовых унифицированных submodule: когнитивного, приводного и обобщённого сенсорного submodule. Наличие таких унифицированных submodule *облегчает сборку различных транспортных платформ*.

Были изготовлены прототипы submodule для первого уровня иерархии: когнитивный submodule, submodule ДПТ и силовой модуль. Для проведения тестов также были разработаны и изготовлены базовые submodule, позволяющие отладить работу протокола. Экспериментальные исследования, выполненные на изготовленных submodule, показали, что при подключении submodule с уникальными идентификаторами к шине система может начать *работу в режиме plug-and-play*, т.е. без дополнительных настроек.

Разработан протокол межмодульного взаимодействия, использующий минимальный набор типовых сообщений, что позволило использовать CAN-шину для обеспечения функционирования субмодулей *в режиме реального времени*. Эксперимент с тремя субмодулями показал способность протокола работать в режиме реального времени: задержка, вызывающая позиционную ошибку транспортной платформы в 0.8–10 см, не должна сказываться на работе мобильного робота при использовании дистанционных сенсоров.

Предложенный подход и результаты экспериментального исследования показали, что разделение полнофункционального транспортного модуля на подмодули, основанный на иерархическом принципе построения многоуровневой системы управления модульного мобильного робота, позволяет увеличить быстродействие всей системы за счёт распределения вычислительной нагрузки между модулями и субмодулями при использовании в каждом модуле и субмодуле встраиваемых систем – относительно дешёвых микропроцессоров или одноплатных ЭВМ невысокой производительности.

Заключение. Значительная часть дальнейшей работы будет заключаться в исследовании системы с большим числом субмодулей: определение зависимости задержек приёма сообщений и скорости их передачи от числа подписчиков, величины колебаний задержек при передаче пакетов и т.д. Необходимость этого обусловлена тем, что в одной транспортной платформе может быть до 10 субмодулей, активно публикующих данные с частотой 100 Гц и больше.

Важной задачей является изготовление других субмодулей (например, обобщённого сенсорного) и их отладка, что позволит в дальнейшем реализовать макет транспортного модуля, включающего в себя все упомянутые выше блоки.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Phidgets Inc. – Products for USB Sensing and Control. – URL: <https://www.phidgets.com/> (дата обращения: 26.02.20).
2. HEBI Robotics. – URL: <https://www.hebirobotics.com> (дата обращения: 26.02.20).
3. *Iñigo-Blasco P., Diaz-del Rio F., Romero-Ternero M., Cagigas-Muñiz D. & Vicente-Diaz S.* Robotics software frameworks for multiagent robotic systems development // *Robotics and Autonomous Systems*. – 2012. – Vol. 60, No. 6. – P. 803-821. – ISSN: 0921-8890.
4. *Herbrechtsmeier S., Korthals T., Schopping T. & Ruckert U.* AMiRo: a modular & customizable open-source mini robot platform // 20th International Conference on System Theory, Control and Computing (ICSTCC), Sinaia. 2016. – P. 687-692.
5. *Kalouche S., Rollinson D. & Choset H.* Modularity for maximum mobility and manipulation: control of a reconfigurable legged robot with series-elastic actuators // *IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*. – West Lafayette, IN, USA, 18-20 Oct. 2015. – P. 1-8. IEEE, New York.
6. *Virk G.* CLAWAR Modularity for Robotic Systems // *The International Journal of Robotics Research*. – March-April 2003. – Vol. 22, No. 3-4. – P. 265-277. – ISSN: 02783649.
7. *Steinbauer G., Wotawa F. & Fraser G.* A modular architecture for a multi-purpose mobile robot, *Innovations in Applied Artificial Intelligence // IEA/AIE 2004. Lecture Notes in Computer Science*. – Springer, Berlin, Heidelberg, 2004. – P. 1007-1015. – ISBN 978-3-540-24677-0. – DOI: 10.1007/978-3-540-24677-0_103.
8. *Taira T., Kamata N. & Yamasaki N.* Design and implementation of reconfigurable modular humanoid robot architecture // *IEEE/RSJ International Conference on Intelligent Robots and Systems 2005*, 2-6 Aug. 2005, Edmonton, Alta., Canada. IEEE, New York. – ISSN 2153-0858. – P. 1071-1076. – DOI: 10.1109/IROS.2005.1545122.
9. *Ahn H., Beak Y., Sa I., Kang W., Na J. and Choi J.* Design of reconfigurable heterogeneous modular architecture for service robots // *IEEE/RSJ International Conference on Intelligent Robots and Systems 2008*, 22-26 Sept. 2008, Nice, France. IEEE, New York. – P. 1313-1318. – ISSN 2153-0858. – ISBN 978-1-4244-2057-5. – DOI: 10.1109/IROS.2008.4650706.

10. *Hild M., Siedel T., Benckendorff C., Thiele C. & Spranger M.* Myon, a New Humanoid, In: Language Grounding in Robots, Steels L., Hild M. (eds). – Springer, 2012. – P. 25-44. – ISBN 978-1-4614-3063-6. Boston, MA.
11. *Roh S., Yang K., Park J., Moon H., Kim H.-S., Lee H. & Choi H.* A modularized personal robot DRP I: design and implementation // IEEE Transactions on Robotics. – 2009. – Vol. 25, No. 2. – P. 414-425. – ISSN 1552-3098. – DOI: 10.1109/TRO.2009.2014499.
12. *Magnenat S., Rétornaz P., Bonani M., Longchamp V. & Mondada F.* ASEBA: a modular architecture for event based control of complex robots // IEEE/ASME Transactions on Mechatronics. – 2010. – Vol. 16, No. 2. – P. 321-329. – ISSN 1083-4435. – DOI: 10.1109/TMECH.2010.2042722.
13. *Bonarini A., Matteucci M., Migliavacca M. & Rizzi D.* R2P: An open source hardware and software modular approach to robot prototyping // Robotics and Autonomous Systems. – 2014. – Vol. 62, Issue 7. – P. 1073-1084. – ISSN 0921-8890, DOI: 10.1016/j.robot.2013.08.009.
14. *Losada, D.P., Fernández J.L.; Paz E. & Rafael S.* Distributed and modular CAN-based architecture for hardware control and sensor data integration // Sensors. – 2017. – Vol. 17 (5). – P. 1013-1030. – ISSN 1424-8220. – DOI: 10.3390/s17051013.
15. *Shmakov O., Korolev D., Popov D., Kitaev N. & Korotkov A.* Modular Mobile Robotic Kit for Prototyping and Debugging of Control Algorithms // Proceedings of the 28th DAAAM International Symposium. – 2017. – P. 0950-0956.
16. *Mayoral V., Hernandez A., Kojcev R., Muguruza I. Zamalloa I. Bilbao A. & Usategi L.* The shift in the robotics paradigm – the Hardware Robot Operating System (H-ROS); an infrastructure to create interoperable robot components, NASA/ESA Conference on Adaptive Hardware and Systems (AHS) 2017, Pasadena, CA, USA, 24-27 July 2017. – P. 229-236. – ISSN 2471-769X, ISBN 978-1-5386-3439-4. IEEE, New York. – DOI: 10.1109/AHS.2017.8046383.
17. *Jahn U. Wolff C. & Schulz P.* Concepts of a modular system architecture for distributed robotic systems // Journal of Computers. – 2019. – Vol. 8, Issue 1. – 16 p. – ISSN 2073-431X. – DOI: 10.3390/computers8010025.
18. *Andreev V.P.* The Concept of Using the Theory of Multi-Agent Systems to Design Control Systems for Mobile Robots with Modular Architecture // Proceedings of the International Scientific and Technological Conference EXTREME ROBOTICS-2019. – SaintPetersburg: OOO "Izdatel'sko-poligraficheskii kompleks "Gangut" Publ., 2019. – P. 524-534.
19. *Андреев В.П., Ким В.Л., Плетенев П.Ф.* Принцип полной функциональности модулей в гетерогенных модульных мобильных роботах // Экстремальная робототехника (ЭР-2017): Тр. международной научно-технической конференции. – СПб.: ЦНИИ РТК, 2017. – № 1. – С. 81-91.
20. *Андреев В.П., Ким В.Л.* Разработка функциональных узлов гетерогенного модульного мобильного робота // Экстремальная робототехника (ЭР-2016): Тр. международной научно-технической конференции. – СПб.: ООО «АП4Принт», 2016. – С. 359-369.
21. *Юревич Е.И.* Основы робототехники: учебник для вузов. – Л.: Изд-во Машиностроение, Ленингр. отд-ние, 1985. – 271 с.
22. *Siciliano B. Sciacivico L. [et al.].* Robotics Modelling, Planning and Control. – Springer-Verlag London Limited, 2009. – 632 p.
23. *Siegwart R., Nourbakhsh I., Scaramuzza D.* Introduction to Autonomous Mobile Robots. – Cambridge: The MIT Press, 2004. – 472 p.
24. Технические характеристики сервисного робота Care-O-bot 4. – URL: <https://www.care-obot.de/de/care-o-bot-4/technical-data.html/> (дата обращения: 26.02.20).
25. Kiva Systems: Three Engineers, Hundreds of Robots, One Warehouse. – URL: <https://spectrum.ieee.org/robotics/robotics-software/three-engineers-hundreds-of-robots-one-warehouse> (дата обращения: 26.02.20).
26. Сервисный гуманоидный робот Rollin' Justin, – URL: <https://www.dlr.de/rm/en/desktopdefault.aspx/tabid-11427/#gallery/29202> (дата обращения: 26.02.20).

REFERENCES

1. Phidgets Inc. Products for USB Sensing and Control. Available at: <https://www.phidgets.com/> (accessed 26 February 20).
2. HEBI Robotics. Available at: <https://www.hebirobotics.com> (accessed 26 February 20).

3. Iñigo-Blasco P., Diaz-del Rio F., Romero-Ternero M., Cagigas-Muñiz D. & Vicente-Diaz S. Robotics software frameworks for multiagent robotic systems development, *Robotics and Autonomous Systems*, 2012, Vol. 60, No. 6, pp. 803-821. ISSN: 0921-8890.
4. Herbrechtsmeier S., Korthals T., Schopping T. & Ruckert U. AMiRo: a modular & customizable open-source mini robot platform, *20th International Conference on System Theory, Control and Computing (ICSTCC), Sinaia. 2016*, pp. 687-692.
5. Kalouche S., Rollinson D. & Choset H. Modularity for maximum mobility and manipulation: control of a reconfigurable legged robot with series-elastic actuators, *IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*. West Lafayette, IN, USA, 18-20 Oct. 2015, pp. 1-8. IEEE, New York.
6. Virk G. CLAWAR Modularity for Robotic Systems, *The International Journal of Robotics Research*, March-April 2003, Vol. 22, No. 3-4, pp. 265-277. ISSN: 02783649.
7. Steinbauer G., Wotawa F. & Fraser G. A modular architecture for a multi-purpose mobile robot, *Innovations in Applied Artificial Intelligence, IEA/AIE 2004. Lecture Notes in Computer Science*, pp. 1007-1015. Springer, Berlin, Heidelberg. ISBN 978-3-540-24677-0. DOI: 10.1007/978-3-540-24677-0_103.
8. Taira T., Kamata N. & Yamasaki N. Design and implementation of reconfigurable modular humanoid robot architecture, *IEEE/RSJ International Conference on Intelligent Robots and Systems 2005, 2-6 Aug. 2005, Edmonton, Alta., Canada*. IEEE, New York. ISSN 2153-0858, pp. 1071-1076. DOI: 10.1109/IROS.2005.1545122.
9. Ahn H., Beak Y., Sa I., Kang W., Na J. and Choi J. Design of reconfigurable heterogeneous modular architecture for service robots, *IEEE/RSJ International Conference on Intelligent Robots and Systems 2008, 22-26 Sept. 2008, Nice, France*, pp. 1313-1318, IEEE, New York. ISSN 2153-0858. ISBN 978-1-4244-2057-5. DOI: 10.1109/IROS.2008.4650706.
10. Hild M., Siedel T., Benckendorff C., Thiele C. & Spranger M. Myon, a New Humanoid, In: *Language Grounding in Robots*, Steels L., Hild M. (eds). Springer, 2012, pp. 25-44. ISBN 978-1-4614-3063-6. Boston, MA.
11. Roh S., Yang K., Park J., Moon H., Kim H.-S., Lee H. & Choi H. A modularized personal robot DRP I: design and implementation, *IEEE Transactions on Robotics*, 2009, Vol. 25, No. 2, pp. 414-425. ISSN 1552-3098. DOI: 10.1109/TRO.2009.2014499.
12. Magnenat S., Rétornaz P., Bonani M., Longchamp V. & Mondada F. ASEBA: a modular architecture for event based control of complex robots, *IEEE/ASME Transactions on Mechatronics*, 2010, Vol. 16, No. 2, pp. 321-329. ISSN 1083-4435. DOI: 10.1109/TMECH.2010.2042722.
13. Bonarini A., Matteucci M., Migliavacca M. & Rizzi D. R2P: An open source hardware and software modular approach to robot prototyping, *Robotics and Autonomous Systems*, 2014, Vol. 62, Issue 7, pp. 1073-1084. ISSN 0921-8890, DOI: 10.1016/j.robot.2013.08.009.
14. Losada, D.P., Fernández J.L.; Paz E. & Rafael S. Distributed and modular CAN-based architecture for hardware control and sensor data integration, *Sensors*, 2017, Vol. 17 (5), pp. 1013-1030. ISSN 1424-8220. DOI: 10.3390/s17051013.
15. Shmakov O., Korolev D., Popov D., Kitaev N. & Korotkov A. Modular Mobile Robotic Kit for Prototyping and Debugging of Control Algorithms, *Proceedings of the 28th DAAAM International Symposium*, 2017, pp. 0950-0956.
16. Mayoral V., Hernandez A., Kojcev R., Muguruza I. Zamalloa I. Bilbao A. & Usategi L. The shift in the robotics paradigm – the Hardware Robot Operating System (H-ROS); an infrastructure to create interoperable robot components, *NASA/ESA Conference on Adaptive Hardware and Systems (AHS) 2017, Pasadena, CA, USA, 24-27 July 2017*, pp. 229-236. ISSN 2471-769X, ISBN 978-1-5386-3439-4. IEEE, New York, DOI: 10.1109/AHS.2017.8046383.
17. Jahn U. Wolff C. & Schulz P. Concepts of a modular system architecture for distributed robotic systems, *Journal of Computers*, 2019, Vol. 8, Issue 1, 16 p. ISSN 2073-431X. DOI: 10.3390/computers8010025.
18. Andreev V.P. The Concept of Using the Theory of Multi-Agent Systems to Design Control Systems for Mobile Robots with Modular Architecture, *Proceedings of the International Scientific and Technological Conference EXTREME ROBOTICS-2019*. SaintPetersburg: OOO "Izdatel'sko-poligraficheskii kompleks "Gangut" Publ., 2019, pp. 524-534.
19. Andreev V.P., Kim V.L., Pletenev P.F. Printsip polnoy funktsional'nosti moduley v geterogennykh modul'nykh mobil'nykh robotakh [The principle of full functionality of modules in heterogeneous modular mobile robots], *Ekstremal'naya robototekhnika (ER-2017): Tr. mezhdunarodnoy nauchno-tekhnicheskoy konferentsii [Extreme robotics (ER-2017): Proceedings of the international scientific and technical conference]*. Saint Peyersburg: TSNII RTK, 2017, No. 1, pp. 81-91.

20. *Andreev V.P., Kim V.L.* Razrabotka funktsional'nykh uzlov geterogenного modul'nogo mobil'nogo robota [Development of functional nodes of a heterogeneous modular mobile robot], *Ekstremal'naya robototekhnika (ER-2017): Tr. mezhdunarodnoy nauchno-tekhnicheskoй konferentsii* [Extreme robotics (ER-2017): Proceedings of the international scientific and technical conference]. Saint Peetersburg: ООО «AP4Print», 2016, pp. 359-369.
21. *Yurevich E.I.* Osnovy robototekhniki: uchebник dlya vtuzov [Fundamentals of robotics: a textbook for higher education institutions]. Leningrad: Izd-vo Mashinostroenie, Leningr. otd-nie, 1985, 271 p.
22. *Siciliano B. Sciacivco L. [et al.]*. Robotics Modelling, Planning and Control. Springer-Verlag London Limited, 2009, 632 p.
23. *Siegwart R., Nourbakhsh I., Scaramuzza D.* Introduction to Autonomous Mobile Robots. Cambridge: The MIT Press, 2004, 472 p.
24. Tekhnicheskie kharakteristiki servisnogo robota Care-O-bot 4 [Technical characteristics of the Care-O-bot 4 service robot]. Available at: <https://www.care-o-bot.de/de/care-o-bot-4/technical-data.html/> (accessed 26 February 20).
25. Kiva Systems: Three Engineers, Hundreds of Robots, One Warehouse. Available at: <https://spectrum.ieee.org/robotics/robotics-software/three-engineers-hundreds-of-robots-one-warehouse> (accessed 26 February 20).
26. Servisnyy gumanoidnyy robot Rollin' Justin [Service humanoid robot Rollin' Justin]. Available at: <https://www.dlr.de/rm/en/desktopdefault.aspx/tabid-11427/#gallery/29202> (accessed 26 February 20).

Статью рекомендовал к опубликованию д.т.н. Ю.В. Подураев.

Андреев Виктор Павлович – Московский государственный технологический университет «СТАНКИН» (МГТУ "СТАНКИН"); e-mail: andreevvipa@yandex.ru; 127055, Москва, Вадковский пер., 1; тел.: +79652107951; д.т.н.; к.ф.-м.н.; профессор; кафедра сенсорных и управляющих систем.

Ким Валерий Леонидович – e-mail: top7733@gmail.com; тел.: +79153041436; кафедра робототехники и мехатроники; аспирант; ассистент.

Эприков Станислав Робертович – Институт прикладной математики им М.В. Келдыша Российской академии наук; e-mail: stanislav.eprikov@gmail.com; 125047, г. Москва, Миусская пл., 4; тел.: +79175254747; аспирант.

Andreev Victor Pavlovich – Moscow State University of Technology «STANKIN» (MSTU «STANKIN»); e-mail: andreevvipa@yandex.ru; 1, Vadkovsky per., Moscow, 127055, Russia; phone: +79652107951; the department of sensory and drive systems; dr. of eng. sc.; professor .

Kim Valerii Leonidovich – e-mail: top7733@gmail.com; phone: +79153041436; the department of robotics and mechatronics; postgraduate student; assistant.

Eprikov Stanislav Robertovich – The Keldysh Institute of Applied Mathematics; e-mail: stanislav.eprikov@gmail.com; 4, Miusskya pl., Moscow, 125047, Russia; phone: +79175254747; postgraduate student.

УДК 004.932.2

DOI 10.18522/2311-3103-2020-1-218-232

**В.А. Бондаренко, Д.К. Ельцова, А.И. Лизин, В.А. Павлова, М.В. Созинова,
В.А. Тупиков**

МНОГОАГЕНТНЫЙ АЛГОРИТМ АВТОМАТИЧЕСКОГО ОБНАРУЖЕНИЯ И СОПРОВОЖДЕНИЯ НЕДЕТЕРМИНИРОВАННЫХ ОБЪЕКТОВ

В целях разработки робастного алгоритма автоматического обнаружения и сопровождения недетерминированных объектов для встраиваемых вычислительных систем с оптико-электронными устройствами. В рамках данной работы произведено исследование и анализ имеющегося мирового научно-технического опыта в области алгоритмов авто-