УДК 004.896
DOI: 10.15827/0236-235X.130.243-249
Дата подачи статьи: 13.12.19
2020. Т. 33. № 2. С. 243–249

Разработка системы управления педипуляторами антропоморфного робота АР-601М

М.В. Тарачков [1], *аспирант, mishklgpmi@mail.ru*
О.В. Толстель [1], *к.т.н., доцент, tolstel.oleg@mail.ru*
А.Л. Калабин [2], *д.ф.-м.н., профессор, akalabin@yandex.ru*

[1] *Балтийский федеральный университет им. Иммануила Канта, Институт физико-математических наук и информационных технологий, г. Калининград, 236002, Россия*
[2] *Тверской государственный технический университет, г. Тверь, 170026, Россия*

В статье представлена реализация системы управления педипуляторами для антропоморфного робота АР-601М. Предложенная система управления состоит из драйвера, обеспечивающего взаимодействие бортового компьютера с главным микроконтроллером робота, и управляющей программы, решающей задачу обратной кинематики и позволяющей осуществлять планирование движений каждого педипулятора в заданную точку. Также управляющая программа обладает графическим интерфейсом для отображения трехмерной модели робота в пространстве.

Для построения данной системы управления педипуляторами была использована свободно распространяемая программная платформа Robot Operating System (ROS), в частности, пакеты ROS_Control для реализации низкоуровневого взаимодействия, MoveIt! для планирования движений, RViz для визуализации. Вследствие этого необходима операционная система Linux Ubuntu 16.04. Комплекс программ для управления педипуляторами написан на языке программирования C++. Выбор языка программирования обусловлен тем, что для решения данной задачи необходимы наилучшее быстродействие и соизмеримый с этим объем затрат на написание кода. Для языка C++ существует библиотека ввода-вывода ASIO, при помощи которой и выполняется взаимодействие с контроллером робота.

В статье приведены результаты эксперимента по использованию предложенной системы управления педипуляторами антропоморфного робота АР-601М, а также возможные варианты применения описанной программы.

Разработанная система управления педипуляторами создает уровень абстракции между аппаратным и программным обеспечением, что позволяет исследователям сконцентрироваться на решении задачи хождения. Кроме того, программа обладает удобным графическим визуализатором.

Рассматривается возможность доработки программы для использования в ее составе инерциально-измерительного модуля, который позволит получать данные об ускорениях и угловых скоростях робота, а после применения к ним фильтра Калмана или Маджвика – углы ориентации в пространстве.

Ключевые слова: *антропоморфные роботы, система управления, хождение, robot operating system, ROS_Control, обратная кинематика.*

Создание системы управления педипуляторами в целом является известной задачей. Наиболее важно для ее разработки правильно подобрать инструментарий и иметь техническую документацию от производителя робота. В случае с АР-601М документация была предоставлена НПО «Андроидная техника», также имелась возможность задавать уточняющие вопросы специалистам компании. Известно, что ПО для подобных систем управления уже не раз реализовалось, поэтому грамотным решением является использование готового свободно распространяемого ПО для взаимодействия с оборудованием, для решения задачи обратной кинематики, а также для визуализации процесса планирования движений. Одним из таких инструментов является ROS – свободное ПО, работающее под управлением Linux Ubuntu разных версий и реализующее механизм взаимодействия разнородных программ через обмен сообщениями. Несомненным удобством является то, что ROS может работать на нескольких компьютерах и программы могут обнаруживать друг друга, находясь на разных устройствах. Кроме того, ROS обладает активно развивающимся и расширяющимся сообществом, что, во-первых, обеспечивает поддержку при работе с данной программной платформой, а во-вторых, позволяет использовать наработки других программистов, находящиеся в открытом доступе. ROS обладает качественной документацией. Для реализации

предложенной в статье системы управления ногами антропоморфного робота были доработаны пакеты ROS_Control (низкоуровневое взаимодействие) и MoveIt! (планирование движений).

Модель робота AP-601M в ROS. Еще одним преимуществом ROS является удобный способ описания конструкции робота – формат Unified Robot Description Format (URDF). Согласно ему, робот составляется из сочленений и шарниров (рис. 1).

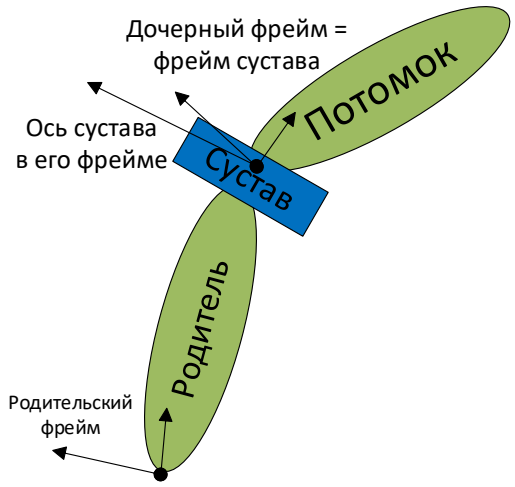

Рис. 1. Представление робота в формате URDF: сочленения, шарниры и их свойства

Fig. 1. The robot representation in URDF format: joints, hinges and their properties

Сочленения описываются при помощи трехмерной модели в формате DAE или STL. Разделяют визуальную часть, используемую для отображения, и часть для определения столкновений. Шарниры описывают способ соединения сочленений. Указываются ось, вокруг которой будет производиться вращение, характер этого вращения (постоянный, в пределах заданного диапазона, вдоль одной оси), ограничения в виде минимального и максимального углов, моментов, скоростей и пр. [1].

Построение модели в САПР SolidWorks. Трехмерные модели сочленений робота были созданы в САПР SolidWorks как в наиболее качественном продукте для трехмерного моделирования в области машиностроения (для нее реализован программный модуль для экспорта модели в формат URDF – SolidWorks To URDF Exporter) [2].

При создании модели сочленения необходимо задать единицы измерения в метрах и следить за расположением системы координат: Z – вверх, X – вперед. Данные требования связаны с тем, что программный пакет для визуализации RViz использует метры в качестве единицы измерения и создание модели, например в миллиметрах, может привести к неточностям. Направление осей Z и X выбрано в ROS как стандартное.

Модели сочленений создаются по отдельности и объединяются в сборку. В каждой модели необходимо создать локальную систему координат и указать ось, относительно которой будет вращаться последующее сочленение. Эти параметры в дальнейшем указываются в URDF Exporter для корректной работы. Также в модели можно создать цилиндрические выступы, которые позволят объединить модели в сборку и провести ось вращения.

Создаются две версии моделей сочленений: одна для визуального отображения (ее следует делать максимально качественной), другая – для расчета столкновений с объектами (должна быть максимально упрощенной). Поскольку алгоритм проверки на столкновения применяется к каждому треугольнику полигональной сетки, то, чем их будет меньше, тем быстрее выполнится работа. Результат построения модели в программе SolidWorks приведен на рисунке 2.

Особенности экспорта в URDF при помощи плагина SolidWorks To URDF Exporter. Программа SolidWorks To URDF Exporter не-

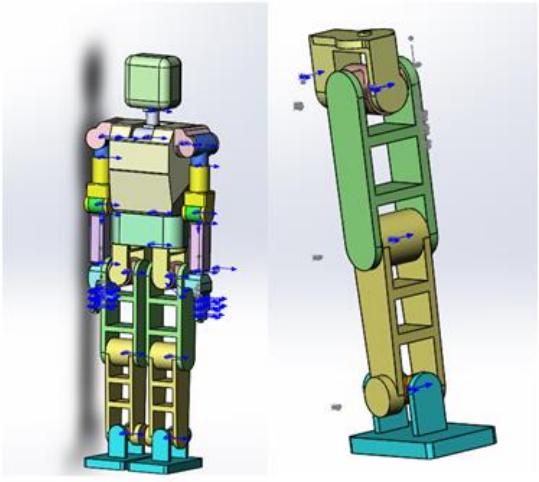

Рис. 2. Сборка модели робота в САПР SolidWorks

Fig. 2. The robot model assembling in CAD SolidWorks

корректно создает пакет для системы ROS, поэтому в него необходимо внести ряд изменений:

– указать правильный путь до файла с URDF-описанием;

– изменить загрузку URDF-описания в launch-файле [3];

– в URDF-описание добавить глобальное сочленение, относительно которого будет считаться расположение всех остальных сочленений.

Пакет ROS_Control. В роботе АР-600 установлен микроконтроллер STM32F437. Это базовый контроллер. Через интерфейс RS-485 он связан с микроконтроллерами STM32F103, расположенными около каждого мотора. Они выдают управляющее воздействие на мотор на основе сигнала обратной связи с инкрементного энкодера. Для контроля управляющего воздействия данный контроллер использует реализованный в нем ПИД-регулятор. Базовый контроллер может через протокол взаимодействия задать коэффициенты регулятора, минимальный и максимальный углы поворота, центровку и прочие параметры. Также базовый контроллер осуществляет контроль напряжений и токов по всем линиям питания (6, 8, 12 и 48 В).

Взаимодействие базового контроллера и управляющего компьютера осуществляется через интерфейс Ethernet (разработка НПО «Андроидная техника») посредством протокола, который работает поверх сетевого протокола UDP.

Согласно протоколу (см. таблицу), на базовый контроллер со стороны управляющего компьютера должна быть отправлена последовательность из 1 472 байтов. На каждый из моторов отведено 16 байтов. Каждый байт описывает определенный параметр.

Программа, осуществляющая взаимодействие с контроллером робота на управляющем компьютере, будет реализована на основе пакета ROS_Control. Этот набор программ создан для обработки сырых данных, поступающих через низкоуровневые интерфейсы, и для передачи их другим узлам ROS в подготовленном для обработки виде. Данные программы реализованы потокобезопасно, а часть из них (контроллеры) могут быть заменены в процессе работы (это может понадобиться, если нужно перейти от управления по моменту к управлению по позиции) [4]. Схема ROS_Control приведена на рисунке 3.

Непосредственно взаимодействие с контроллером робота осуществляет программа для аппаратного взаимодействия с роботом. В ней создается сокет для обмена сообщениями по протоколу UDP. Принятые сырые данные готовятся следующим образом: из 8-битного представления они преобразуются в 16-битные значения, затем в радианы (так как это базовая единица измерения угла в ROS) и наконец в традиционный вид из байтового представления напряжения и тока.

Программы, которые принимают уставку положения шарниров от узлов системы ROS, осуществляют регулирование управляющего

Описание протокола обмена

The exchange protocol description

№ байта	Буфер приема	Буфер передачи
0	Номер мотора	Номер мотора
1	Статус	Вкл./Откл.
2	Текущая позиция (байт 1)	Уставка позиции (байт 1)
3	Текущая позиция (байт 2)	Уставка позиции (байт 2)
4	Напряжение (байт 1)	Уставка коэф. Д (байт 1)
5	Напряжение (байт 2)	Уставка коэф. Д (байт 2)
6	Сила тока (байт 1)	Уставка центровки (байт 1)
7	Сила тока (байт 2)	Уставка центровки (байт 2)
8	Коэф. П (байт 1)	Уставка коэф. П (байт 1)
9	Коэф. П (байт 2)	Уставка коэф. П (байт 2)
10	Коэф. И (байт 1)	Уставка коэф. И (байт 1)
11	Коэф. И (байт 2)	Уставка коэф. И (байт 2)
12	Текущая мин. позиция (байт 1)	Уставка мин. позиции (байт 1)
13	Текущая мин. позиция (байт 2)	Уставка мин. позиции (байт 2)
14	Текущая макс. позиция (байт 1)	Уставка макс. позиции (байт 1)
15	Текущая макс. позиция (байт 2)	Уставка макс. позиции (байт 2)

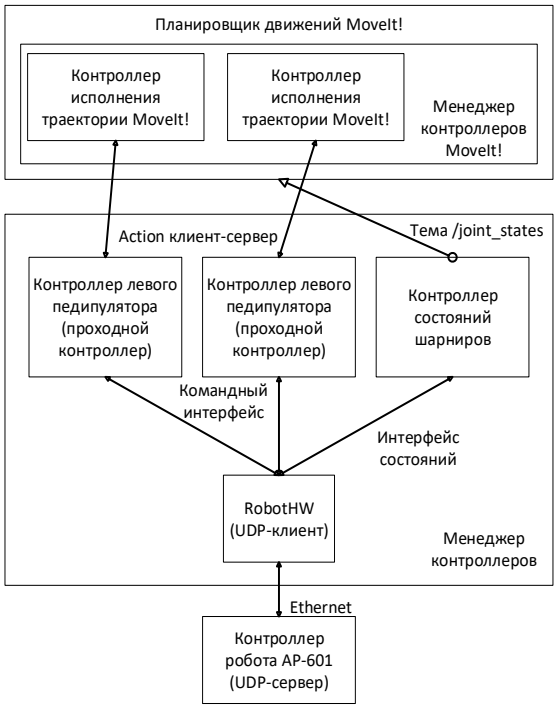

Рис. 3. Общая схема взаимодействия программ в пакетах ROS_Control и MoveIt!

Fig. 3. Interaction general scheme between programs in the ROS_Control and MoveIt! packages

воздействия, а также используются для передачи сигналов обратной связи от робота в систему ROS, называются контроллерами [5].

Контроллеры и программа для аппаратного взаимодействия с роботом связываются друг с другом при помощи интерфейсов, обмен данными происходит каждый квант времени, определенный с помощью параметра задержки.

Контроллеры и другие узлы системы ROS взаимодействуют между собой при помощи сервера действий [6]. Такой сервер позволяет отправлять на контроллер уставку и контролировать процесс ее достижения с помощью обратной связи.

Управление контроллерами (запуск, отключение, замену) осуществляет менеджер контроллеров [7].

Для передачи параметров робота в систему ROS был создан контроллер состояний шарниров.

Для передачи управляющего воздействия на шарниры манипулятора робота и его захватывающее устройство были созданы контроллер исполнения траектории манипулятора и контроллер захватывающего устройства.

Конфигурация робота для MoveIt!. MoveIt! – это пакет в системе ROS, предназначенный для планирования движений. В его состав входят подпрограммы для решения задач прямой и обратной кинематики. MoveIt! имеет удобное API, которое можно использовать на языках C++ и Python. Также есть графический интерфейс для программы RViz [1].

В MoveIt! передаются точка в пространстве, которую нужно переместить в исполнительный элемент педипулятора, конфигурация педипулятора, текущие положения узлов и карта препятствий, позволяющая избежать соударений при движении. Данная программа рассчитывает траекторию, состоящую из последовательных положений шарниров, пройдя которую, манипулятор сможет достичь заданной точки.

Для получения более гладкой траектории полученные данные из программы MoveIt! интерполируют с помощью сплайнов. Это происходит в контроллере исполнения траектории.

Чтобы использовать программу MoveIt! для робота AP-601M, необходимо создать набор конфигурационных файлов. Автоматизировать этот процесс может программа MoveIt Setup Assistant [8].

В данную программу необходимо загрузить URDF-описание робота и выполнить следующие шаги конфигурации.

1. *Создание матриц коллизий.* Они показывают, какие сочленения при движении никогда не могут соприкоснуться друг с другом, например, предплечье и большой палец. Данная матрица позволяет не учитывать такие сочленения при планировании движений, тем самым ускоряя расчеты. Для заполнения матрицы генерируются от 10 до 100 тысяч произвольных позиций всех шарниров. Если сочленения ни разу не пересеклись, они помечаются меткой Never in collision.

2. *Создание групп планирования.* Сочленения и шарниры объединяются в группы планирования. Например, группа планирования педипулятора начинается от шарнира вращения бедра и заканчивается в стопе. Удобно задавать в этом случае группу планирования в виде кинематической цепи (последовательности сочленений). Для группы планирования выбирается программа-решатель задачи обратной кинематики. В текущей конфигурации робота выбраны две группы планирования: шарниры левой ноги, шарниры правой ноги.

3. *Создание начальных положений для групп планирования.* Это набор значений поло-

жений шарниров для некоторых стандартных положений робота. В данной конфигурации выбрано одно начальное положение, когда руки робота опущены вниз.

4. *Указание рабочей точки педипулятора.* Это конечная точка педипулятора, положение которой задается при планировании движений, как правило, последнее сочленение педипулятора – стопа.

Для осуществления взаимодействия с контроллерами робота необходимо создать контроллеры MoveIt!. Они являются клиентами действий [9] и получают данные из MoveIt!. Для управления контроллерами используется диспетчер контроллеров MoveIt!.

Проведение калибровки. Перед использованием манипулятора робота необходимо произвести установку минимальных и максимальных значений положений его шарниров, а также параметров ПИД-регулятора.

Предлагается следующий подход. При отключенном питании и установленной в 0 центровке мотора шарнир устанавливается в нулевое положение. Оно должно совпадать с нулевым положением модели. Центровка мотора устанавливается равной текущему значению положения мотора. Тогда после компиляции и повторного запуска управляющей программы текущее положение шарнира будет равно 0. Далее необходимо отвести шарнир в минимальное, а затем максимальное положение, запомнив в обоих случаях показания энкодеров. Полученные показания указываются в настройках шарниров. После этого выполняется перемещение шарнира в указанное положение в автоматическом режиме. Если у мотора, вращающего шарнир, не хватает мощности, необходимо увеличить коэффициент П регулятора. В случае перерегулирования или большого времени переходного процесса необходимо настроить коэффициенты И и Д.

Полученные в ходе калибровки минимальные и максимальные значения должны быть перенесены в URDF-описание робота.

Модель робота загружается в программу RViz [10], где визуально можно оценить совпадения движений реального робота и модели. Если шарнир вращается в противоположном направлении, то необходимо изменить знак в теге axis на противоположный.

Испытания системы управления. Антропоморфный робот AP-601M был подвешен на кране, к его бортовому компьютеру по интерфейсу Ethernet был подключен управляющий компьютер (рис. 4).

Рис. 4. Расположение робота на кране

Fig. 4. Location work on the crane

В ходе испытаний требовалось переместить педипуляторы робота в произвольное положение. Визуально сравнивалось положение педипуляторов на модели робота (см. http://www.swsys.ru/uploaded/image/2020-2/2020-2-dop/5.jpg) и у реального робота (см. http://www.swsys.ru/uploaded/image/2020-2/2020-2-dop/6.jpg). Проверялась отказоустойчивость системы.

Результаты. В ходе испытаний системы управления педипулятора антропоморфного робота AP-601M было выявлено, что в целом система управления работает стабильно и перемещает педипуляторы в заданное положение.

Полученный результат следует признать удовлетворительным. В ходе работ было выявлено, что не хватает системы отображения данных о состоянии электродвигателей. Также необходимо доработать API MoveIt! путем написания обертки для того, чтобы в будущем ее было проще использовать исследователям.

Заключение. Была разработана система управления педипуляторами антропоморфного робота AP-601M с использованием программной платформы ROS. Система испытана и показала удовлетворительные результаты. Также были выявлены недостатки, которые планируется устранить.

Разработанная система управления может быть использована исследователями для взаимодействия с педипуляторами робота и про-

верки алгоритмов хождения. Планируется доработать систему для поддержки инерциально-навигационного модуля, который при помощи фильтров Калмана [11] или Маджвика [12] позволит получить углы ориентации робота в пространстве.

Литература

1. Joseph L. Mastering ROS for robotics programming. UK, Packt Publ., 2018, 552 c.
2. SolidWorks to URDF Exporter. Wiki.ros.org. URL: http://wiki.ros.org/sw_urdf_exporter (дата обращения: 02.12.2019).
3. Mahtani A., Sanchez L., Fernandez E., Martinez A. Effective robotics programming with ROS. UK, Packt Publ., 2016, 468 p.
4. Тарачков М.В., Ширкин А.Е., Перминов И.К., Письменный В.В., Мыльникова В.А. Разработка элементов системы управления антропоморфным роботом АР-601 // ГИСИС: матер. IV Всерос. конф. Калининград, 2018. С. 368–374.
5. Koubaa A. Robot operating system. The complete reference. Springer Publ., 2016, vol. 1, 720 p.
6. Description of packet Actionlib. Wiki.ros.org. URL: http://wiki.ros.org/actionlib (дата обращения: 02.12.2019).
7. Description of controller manager. Wiki.ros.org. URL: http://wiki.ros.org/controller_manager (дата обращения: 02.12.2019).
8. MoveIt setup assistant. Docs.ros.org. URL: http://docs.ros.org/kinetic/api/moveit_tutorials/html/doc/setup_assistant/setup_assistant_tutorial.html (дата обращения: 02.12.2019).
9. Simple action clients. WIKI.ROS.ORG. URL: http://wiki.ros.org/actionlib_tutorials/Tutorials/SimpleActionClient (дата обращения: 02.12.2019).
10. Visualization program RViz. WIKI.ROS.ORG. URL: http://wiki.ros.org/rviz (дата обращения: 02.12.2019).
11. Kalman filter. Habrahabr. URL: https://habr.com/ru/post/166693/ (дата обращения: 02.12.2019).
12. Madjwick filter. Habrahabr. URL: https://habr.com/ru/post/255661/ (дата обращения: 02.12.2019).

Software & Systems
DOI: 10.15827/0236-235X.130.243-249

Received 13.12.19
2020, vol. 33, no. 2, pp. 243–249

The pedipulator control system development for anthropomorphic robot AR-601M

M.V. Tarachkov [1], *Postgraduate Student, mishklgpmi@mail.ru*
O.V. Tolstel [1], *Ph.D. (Engineering), Associate Professor, tolstel.oleg@mail.ru*
A.L. Kalabin [2], *Dr.Sc. (Physics and Mathematics), Professor, akalabin@yandex.ru*

[1] *Baltic Federal University Immanuel Kant, Institute of Physical and Mathematical Sciences and Information Technologies, 236041, Kaliningrad, Russian Federation*
[2] *Tver State Technical University, Tver, 170026, Russian Federation*

Abstract. The paper presents the pedipulator control system implementation for the anthropomorphic robot AR-601M manufactured by NPO Android Technika LLC (Magnitogorsk, Russia). The proposed control system consists of a driver that provides interaction between the on-board computer and the robot main microcontroller, and a control program that solves the inverse kinematic problem and allows planning the each pedipulator movements to a given point. The control program also has a graphical interface for displaying a three-dimensional model of the robot in space.

To build this pedipulator control system, we used the freely distributed software platform Robot Operating System (ROS) and, in particular, ROS Control packages for implementing low-level interaction, MoveIt! for planning movements, RViz for visualization. Therefore, you must use the Linux Ubuntu 16.04 operating system. A program set for controlling pedipulators is in the C ++ programming language. The programming language choice of is due to the fact that to solve this problem, you need the best performance and a commensurate amount of cost for writing code. For the C ++ language, there is an ASIO input/output library that helps you interact with the robot controller.

There are the experiment results on the proposed control system usage for the anthropomorphic robot AR-601M pedipulators, as well as possible options for using the program described in the paper.

The developed control system for pedipulators creates an abstraction level between hardware and software, which allows researchers to concentrate on solving the walking problem. In addition, the program has a convenient graphical visualizer.

The paper considers the program updating possibility to use an inertial measurement module in its composition, which will allow obtaining data on the accelerations and angular velocities of the robot, and after applying the Kalman or Majvik filter to them, the orientation angles in space.

Keywords: anthropomorphic robots, control system, walking, robot operating system, ROS_Control, inverse kinematics.

References

1. Joseph L. *Mastering ROS for Robotics Programming*. UK, Packt Publ., 2018, 552 p.
2. *SolidWorks to URDF Exporter*. *Wiki.ros.org*. Available at: http://wiki.ros.org/sw_urdf_exporter (accessed December 02, 2019).
3. Mahtani A., Sanchez L., Fernandez E., Martinez A. *Effective Robotics Programming with ROS*. UK, Packt Publ., 2016, 468 p.
4. Tarachkov M.V., Shirkin A.E., Perminov I.K., Pismenny V.V., Mylnikova V.A. Development of elements of the control system anthropomorphic robot AR-601. *Proc. GISIS Conf*., Kaliningrad, 2018, pp. 368–374 (in Russ.).
5. Koubaa A. *Robot Operating System. The Complete Reference*. Springer Publ., 2016, vol. 1, 720 p.
6. *Description of Packet Actionlib. Wiki.ros.org*. Available at: http://wiki.ros.org/actionlib (accessed December 02, 2019).
7. *Description of Controller Manager. Wiki.ros.org*. Available at: http://wiki.ros.org/controller_manager (accessed December 02, 2019).
8. *MoveIt Setup Assistant. Docs.ros.org*. Available at: http://docs.ros.org/kinetic/api/moveit_tutorials/html/doc/setup_assistant/setup_assistant_tutorial.html (accessed December 02, 2019).
9. *Simple Action Clients. WIKI.ROS.ORG*. Available at: http://wiki.ros.org/actionlib_tutorials/Tutorials/SimpleActionClient (accessed December 02, 2019).
10. *Visualization Program RViz. WIKI.ROS.ORG*. Available at: http://wiki.ros.org/rviz (accessed December 02, 2019).
11. *Kalman Filter. Habrahabr*. Available at: https://habr.com/ru/post/166693/ (accessed December 02, 2019).
12. *Madjwick Filter. Habrahabr*. Available at: https://habr.com/ru/post/255661/ (accessed December 02, 2019).

Для цитирования

Тарачков М.В., Толстель О.В., Калабин А.Л. Разработка системы управления педипуляторами антропоморфного робота АР-601М // Программные продукты и системы. 2020. Т. 33. № 2. С. 243–249. DOI: 10.15827/0236-235X.130.243-249.

For citation

Tarachkov M.V., Tolstel O.V., Kalabin A.L. The pedipulator control system development for anthropomorphic robot AR-601M. *Software & Systems*, 2020, vol. 33, no. 2, pp. 243–249 (in Russ.). DOI: 10.15827/0236-235X.130.243-249.