

УДК 681.571

DOI:10.38028/ESI.2022.26.2.015

Разработка библиотеки планирования и управления движениями коллаборативного робота Kuka iiwa

Лаптев Никита Витальевич, Кравченко Андрей Александрович, Гергет Ольга Михайловна

Национальный исследовательский Томский политехнический университет, Россия, Томск, nikitalaptev77@gmail.com

Аннотация. В статье представлена клиент-серверная библиотека для взаимодействия коллаборативного робота Kuka LBR iiwa (на операционной системе KUKA Sunrise.OS) с удаленным персональным компьютером (ПК). Интуитивно понятная высокоуровневая библиотека, реализована в программном пакете Matlab. Включает в себя сервер для контроллера Kuka iiwa, а также клиента на базе среды MathWorks Matlab. Набор инструментов библиотеки включает более 30 функций, охватывающих такие операции как, расчет прямой и обратной кинематики, управление роботом в декартовом пространстве и при помощи наборов изгибов осей, планирование траекторий, графическое отображение и обратная связь. Разработанное программное обеспечение (ПО) работает на удаленном компьютере, подключенном к контроллеру робота через протокол TCP/IP. В статье приведены разработанные методы и возможные примеры управления роботом из библиотеки.

Ключевые слова: Коллаборативный робот, iiwa, KUKA Sunrise.OS, Matlab

Цитирование Лаптев Н.В. Разработка библиотеки планирования и управления движениями коллаборативного робота Kuka iiwa / Н.В. Лаптев, А.А. Кравченко, О.М. Гергет // Информационные и математические технологии в науке и управлении. – 2022. – № 2(26). – С. 161-168. – DOI:10.38028/ESI.2022.26.2.015.

Введение. В настоящее время рынок роботов представлен множеством производителей, каждый из которых предлагает свои роботизированные решения, а также низкоуровневое ПО для управления роботом. Предлагаемое производителями роботов ПО требует знания различных языков программирования, таких как, Java, C++, Python, включая специальные функции робота, что может стать препятствием для многих разработчиков. Еще одним недостатком использования контроллера робота является то, что данные и информация от робота доступны только локально в системе, во время выполнения приложений. Это делает невозможным реализацию приложений с удаленным контролем необходимых для решения задач телемедицины [1], адаптивных роботизированных ячеек [2] и дистанционного управления с применением VR/AR технологий [3]. В связи с этим в рамках работы коллектива исследователей по разработке прототипа роботизированного медицинского инструмента (для репозиции костей таза при нестабильных переломах), была реализована клиент-серверная библиотека планирования и управления движениями робота. Библиотека предназначена для взаимодействия коллаборативного робота Kuka LBR iiwa [4] на операционной системе KUKA Sunrise.OS [5] с удаленным ПК на базе программного пакета Matlab от MathWorks [6].

В робототехнике нередко используется внешний компьютер, работающий вместе с роботом-контроллером, где ПК может использоваться для выполнения вычислительно затратных операций, для обработки данных с датчиков и для реализации алгоритмов управления. В качестве ПО для разработки моделей и алгоритмов управления часто используется программный пакет MATLAB. Этот пакет является важным инженерным инструментом, используемым для разработки высокотехнологичных приложений, и широко используется в образовании, исследованиях и промышленности. Например, в работе [7] реализовано приложение с применением библиотеки Simulink, для реализации контроллера динамической ходьбы для двуногого робота размером с человека. Приложения Kuka Control Toolbox (КСТ) [8] и JOpen-ShowVar [9,10] применяются для управления роботами KUKA с 6 степенями свободы из MATLAB/Java. Однако КСТ и JOpenShowVar поддерживают только промыш-

ленный контроллер KRC [11], поэтому их нельзя использовать для управления роботами KUKA iiwa (поставляемыми с более новым контроллером Sunrise). В статье [12] представлена библиотека KUKA Sunrise Toolbox (KST). Это набор инструментов MATLAB и сервер для Sunrise.OS, который позволяет управлять роботами KUKA iiwa с внешнего компьютера. KST реализует интернет-протокол управления передачей (TCP/IP) для соединения с роботом. Он объединяет несколько функций, включая сетевые функции, функции базового планирования движений, функции управления в режиме реального времени, установки и получения параметров робота, функции общего назначения, функции физического взаимодействия. Однако данная библиотека ограничена в возможности подключения внешней автоматики и инструментов, а также не может планировать сложные траектории из множества различных перемещений. В работе [1] авторы использовали библиотеку ROS [13], для управления роботом iiwa, задачи автономного УЗИ под МРТ-наведением, а в работе [14] авторы представили KUKA-IIWA-API, интерфейс для управления iiwa с помощью ROS. Этот интерфейс обеспечивает функции сбора данных и движения «точка-точка» (PTP), но не поддерживает функцию управления в режиме реального времени.

В результате оценки представленных в открытых источниках приложений для планирования и управления коллаборативных роботов и специфики решаемых научным коллективом задач была разработана собственная библиотека удаленного управления Kuka LBR iiwa. Разработанная библиотека имеет расширенные возможности по управлению роботом в режиме планирования траектории, в том числе и смешивание различных видов движений в один пакет, более современные и быстрые способы коммуникации с роботом со стороны клиента, а также гибкость в работе с портами Ю. Библиотека также позволяет расширять свою функциональность за счет модульной архитектуры.

1. Описание используемых инструментов. KUKA iiwa — это легкий коллаборативный манипулятор с семью осями, каждая ось которого оснащена силомоментным датчиком, а также датчиком абсолютного положения. Данные с осей, а также высокая скорость обновления (с интервалами обновления до миллисекунд) позволяют роботу своевременно реагировать на внешние воздействия, что делает робота подходящим для взаимодействия с людьми без использования дополнительных средств безопасности. Робот KUKA iiwa может быть запрограммирован для решения различных задач с помощью среды KUKA Sunrise.OS. Она включает в себя операционную систему "KUKA Sunrise.OS", которая может выполнять программы, написанные на языке Java, а также контроллер "KUKA Sunrise Cabinet". Хотя язык Java является распространенным языком, для программирования робота и использования его функциональности требуется глубокое знание системы Sunrise и различных методов библиотек (API системы).

Из «коробки» робот может управляться только через программу, запущенную на контроллере, а все данные обратной связи недоступны для чтения или изменения извне системы Sunrise.OS. Несмотря на то, что удаленное управление роботом и данные обратной связи можно передавать на удаленный компьютер с помощью пакета FRI, необходимость в совместимости со вспомогательными системами и программным пакетом Matlab привела нас к разработке собственного решения.

2. Описание библиотеки. Сервер библиотеки написан на языке Java и использует Sunrise.OS API 1.7 для реализации функций библиотеки. Сервер основан на базе асинхронного однопоточного TCP-сокета, и поддерживает работу с несколькими клиентами. На стороне клиента MATLAB используются встроенные синхронные Java-сокеты, обеспечивая задержку при обмене информацией менее 1 мс и общую скорость обмена данными порядка 15 кГц. Тем не менее, для обратной совместимости и для случаев, где нужна наибольшая надежность

работы, предусмотрена возможность соединения с сервером, используя встроенные классы MATLAB, такие, как `tcpclient` и `tcpip`.

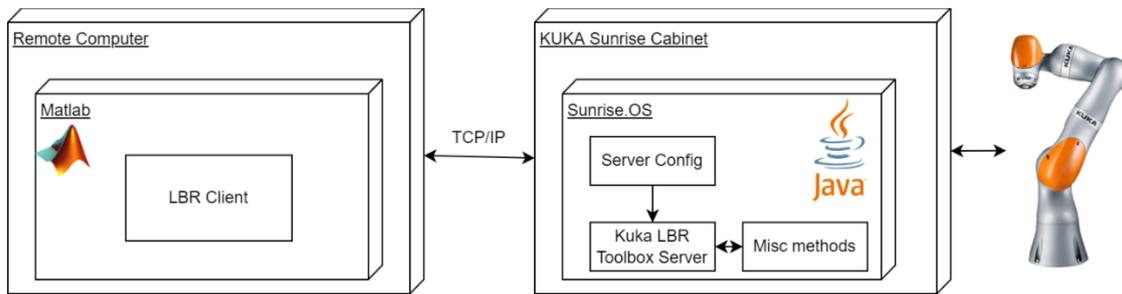


Рис. 1. Архитектура библиотеки, схема связи

Схема связи разработанной библиотеки представлена на рисунке 1, библиотека имеет 35 методов для взаимодействия с роботом. Большинство методов имеют развитую систему входных аргументов, позволяющую гибко настраивать те или иные параметры действия робота. Также библиотека имеет дополнительный класс для описания движения робота, который позволяет задать различные параметры движения, такие, как тип, скорость, ускорение, условия останова и прочее.

Методы логически разделены на 5 главных категорий:

1. Методы получения информации: к данной группе относятся методы обратной связи, позволяющие получать данные по положению робота, его инструмента, данные кинематики или усилий на работе.
2. Методы манипулирования роботом: методы, позволяющие получать или изменять параметры робота. К ним относятся методы работы с инструментами (в частности, программное подключение инструмента, смена его фреймов, получение данных о текущем инструменте и инструментах, доступных на работе), работа с портами ввода-вывода.
3. Методы управления движениями робота: к этой группе относятся команды базового контроля положения (через базовые операции перемещений, таких, как точка-точка или линейное перемещение), а также метод перемещения, принимающий на вход последовательность движений, созданную пользователем с помощью класса описания движений.
4. Методы сетевого контроля: к этой группе относятся группы контроля связи с роботом – замеры скорости реакции контроллера на команды, метод проверки связи.
5. Вспомогательные методы: методы, не взаимодействующие в ходе своей работы с роботом, но предоставляющие пользователю некоторые методы для упрощения программирования под робота.

Для работы с кинематикой были написаны собственные реализации алгоритмов прямой-обратной кинематики, а также добавлена возможность вызова встроенной в Sunrise.OS функции расчёта обратной кинематики. Алгоритм обратной кинематики основан на собственной реализации алгоритма обратной кинематики прямого и обратного следования.

(FABRIK) [15]. Собственная реализация нечувствительна к кинематическим сингулярностям, по сравнению со встроенной в Sunrise.OS API.

Методы движения допускают группировку различных перемещений в одну последовательность, к примеру, одновременное использование перемещений точка-точка (PTP / Point-To-Point), линейное (LIN / Linear), точка-точка в пределах одного пакета. Это позволяет строить сложные траектории и выполнять их в одно движение. Данное действие возможно за счет наследования различных типов движений от единого родительского класса в иерархии

Sunrise.OS, что позволяет впоследствии поместить собранный массив движений в один пакет движений. Подобные пакеты могут содержать до нескольких сотен движений.

В сравнении с Kuka Sunrise Toolbox, библиотека позволяет получать некоторые данные из памяти робота, такие, как данные по имеющимся в базе инструментам, их матрицам трансформации, данные по сохраненным точкам, а также данные по сохраненным в системе объектам и заготовкам. Стоит также отметить, что библиотека не привязана к портам ввода-вывода фланца, и может работать с различными группами Ю, после указания групп в конфигурационном классе сервера.

Библиотека запускается на удаленной системе, взаимодействуя с роботом через протокол TCP/IP, используя порт X66 на контроллере робота. Система поддерживает всю линейку роботов Kuka LBR iiwa. Для работы не требуются дополнительные программные пакеты.

Так как MATLAB является кроссплатформенной средой, библиотека может работать в различных операционных системах, таких, как Windows, Linux и macOS. При этом возможен порт библиотеки в свободную среду GNU Octave или реализация клиентов в других языках и средах программирования.

Исходный код сервера, клиента, а также примеры работы с библиотекой доступны для скачивания на GitHub-странице пакета [16].

3. Примеры использования библиотеки. В качестве примера работы со встроенными функциями прямой-обратной кинематики предлагается рассмотреть код (листинг 1). В примере продемонстрирован расчёт обратной кинематики для робота, в целях описания окружности, после расчета координат робот двигается по полученной траектории, используя набор конфигураций.

Листинг 1. Пример кода

```
% Подключение к роботу
lbr = iiwa("ip", "172.31.1.147");
% Центр окружности в координатах 700, 200, 600 мм
o = [700; 200; 600];
% Фланец робота направлен по оси X перпендикулярно плоскости YZ
orient = deg2rad([0;90;0]);
ang=[];
plt = figure; hold on; axis equal
for i = 1:100
    T = o + [0; 100*sin(2*pi/batchSize*i); 150*cos(2*pi/batchSize*i)];
    % Отображаем целевую координату
    lbr.drawPosition(plt.CurrentAxes, T, "data", "coords", "symbol", 'o');
    % Расчёт обратной кинематики с использованием встроенного алгоритма
    % относительно фланца
    [angles, status] = lbr.getInverseKinematics(T', orient', ...
        "useLocalSolver", true, "toolRelated", false);
    % Если решения существуют - записываем конфигурацию и проверяем её
    % прямой кинематикой
    if status == 0
        res = lbr.getForwardKinematics(angles(1,:), "toolRelated", false, ...
            "useLocalSolver", true);
        % Отображаем координату, полученную через прямую кинематику
        lbr.drawPosition(plt.CurrentAxes, res.Flange, "data", "transMatrix", 'symbol', '+');
        ang(end+1,:)=angles(1,:);
    end
end
% Отображаем конфигурацию робота в последней точке
lbr.drawPosition(plt.CurrentAxes, ang(end,:), "data", "angles", ...
```

```
'symbol', '+', 'drawRobotPose', true);
% Совершаем пакетное движение, в качестве траектории - набор конфигураций
% робота, полученных ранее
[status, msg] = lbr.splineJP(ang);
```

На рисунке **Ошибка! Источник ссылки не найден.** продемонстрирована описываемая роботом траектория, а также конфигурация расположения осей робота, которую он примет в конце движения.

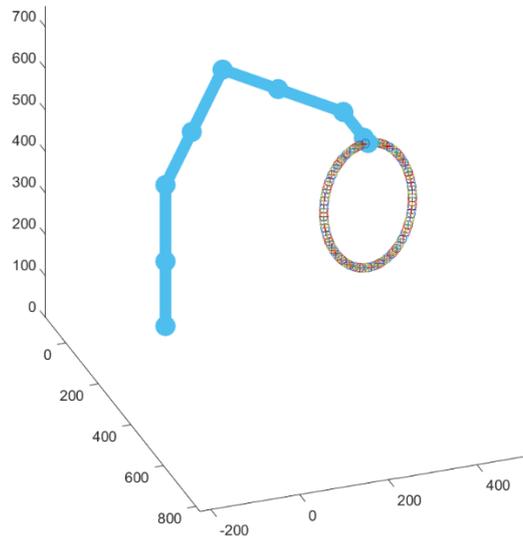


Рис. 2. Визуализация траектории движения и конфигурации робота

Прямая-обратная кинематика может быть использована для предварительного расчёта траектории движения, или создания своей собственной траектории, которую робот выполнит через сглаженное движение по набору конфигураций осей (SplineJP). Эти методы могут быть применимы для расчёта траекторий, обхода коллизий на этапе планирования движений, предварительного расчёта конфигураций робота и для оценки области движения робота.

Ниже приведён пример программного кода, представляющий возможность работы со смешанным типом движений (листинг 2). Пример демонстрирует подключение инструмента, хранящегося в памяти робота, с последующим перемещением по трем точкам, сохраненным в контроллере, в режиме сплайн – линейное перемещение – сплайн.

Листинг 2. Пример кода

```
% Подключение к роботу
lbr = iiwa("ip", "172.31.1.147");
% Получаем список доступных инструментов из системы и подключаем первый
tools = lbr.getToolsAvailable();
lbr.toolAttach(tools(1));
% Получаем список сохраненных точек
points = lbr.getPoints();
% Создаем 3 движения из 3 точек, устанавливаем ограничение по скорости робота
point1 = motion("spl", points(1).coord, points(1).orient);
point1.setCartVel(0.2);
point2 = motion("lin", points(2).coord, points(2).orient);
point2.setCartVel(0.5);
point3 = motion("spl", points(3).coord, points(3).orient);
point3.setCartVel(0.1);
% Совершаем движение в режиме Spline по трем созданным движениям
[status, msg] = lbr.complexMove("spline", [point1 point2 point3]);
% Отсоединяем инструмент
lbr.toolDeattach();
```

При подсоединении инструмента робот передает клиенту координаты и другие параметры по отношению к инструменту, если в методе явно не задано обратное. При создании смешанных типов движений на пакет накладываются ограничения в соответствии с ограничениями Sunrise.OS API. В примере, в режиме сглаживания движений (Spline) невозможно создать движение вида Точка-Точка, а в режиме SplineJP создать траекторию из точек в декартовом пространстве: для этого необходимо задание углов поворота осей.

Таким образом были продемонстрированы примеры взаимодействия с основными функциями библиотеки.

Заключение. В статье представлен набор инструментов MATLAB для общего управления роботом. Обсуждаются основные компоненты и рабочий процесс библиотеки планирования и управления перемещением коллаборативного робота. Представлены примеры, демонстрирующие работу платформы.

Данная библиотека разработана в рамках работы по прототипу роботизированного медицинского инструмента для репозиции костей таза при нестабильных переломах. Библиотека позволяет выполнить полный спектр задач, связанных с манипулированием роботом и его перемещением. Использование библиотеки позволяет писать основной код алгоритмов в среде MATLAB, что упрощает работу для коллектива. За счет расширенных возможностей по работе с Ю-группами и движениями была достигнута требуемая гибкость применения системы.

В рамках развития библиотеки планируется добавление режима управления роботом в реальном времени, а также расширение возможностей по тонкой настройке движений робота.

Благодарности. Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 20-37-90055 «Аспиранты» и государственного задания «Наука» №FFSWW-2020-0014.

Список источников

1. Hennemersperger C. et al. Towards MRI-based autonomous robotic US acquisitions: a first feasibility study, *IEEE Trans. Med. Imaging*, 2016, vol. 36, no. 2, pp. 538–548.
2. Obal P., Gierlak P. EGM Toolbox—Interface for Controlling ABB Robots in Simulink. *Sensors. Multidisciplinary Digital Publishing Institute*, 2021, vol. 21, no. 22, pp. 7463.
3. Ostanin M., Yagfarov R., Klimchik A. Interactive Robots Control Using Mixed Reality. *IFAC-PapersOnLine*, 2019, vol. 52, no. 13, pp. 695–700.
4. LBR iiwa | KUKA in Malaysia. available at: <https://www.kuka.com/en-my/products/robotics-systems/industrial-robots/lbr-iiwa> (accessed: 19.04.2022).
5. KUKA Sunrise.OS | KUKA. available at: <https://www.kuka.com/en-us/products/robotics-systems/software/system-software/sunriseos> (accessed: 04.05.2022).
6. De Filippis L., Guglieri G., Quagliotti F. Path planning strategies for UAVS in 3D environments. *Journal of Intelligent & Robotic Systems, Theory Appl*, 2012, vol. 65, no. 1–4, pp. 247–264.
7. Hubicki C. et al. Walking and running with passive compliance: Lessons from engineering: A live demonstration of the atrias biped. *IEEE Robotics & Automation Magazine*, 2018, vol. 25, no. 3, pp. 23–39.
8. Chinello F. et al. Kuka control toolbox. *IEEE Robotics & Automation Magazine*, 2011, vol. 18, no. 4, pp. 69–79.
9. Sanfilippo F. et al. JOpenShowVar: an open-source cross-platform communication interface to kuka robots. *2014 IEEE International Conference on Information and Automation (ICIA)*. IEEE, 2014, pp. 1154–1159.
10. Sanfilippo F. et al. Controlling Kuka industrial robots: Flexible communication interface JOpenShowVar. *IEEE Robotics & Automation Magazine*, 2015, vol. 22, no. 4, pp. 96–109.
11. KUKA KR C4 | KUKA AG. available at: <https://www.kuka.com/en-us/products/robotics-systems/robot-controllers/kr-c4> (accessed: 04.05.2022).
12. Safeea M., Neto P. Kuka sunrise toolbox: Interfacing collaborative robots with matlab. *IEEE Robotics & Automation Magazine*, 2018, vol. 26, no. 1, pp. 91–96.
13. ROS: Home. available at: <https://www.ros.org/> (accessed: 04.05.2022).

14. Mokaram S. et al. A ROS-integrated API for the KUKA LBR iiwa collaborative robot. IFAC-PapersOnLine. Elsevier, 2017, vol. 50, no. 1, pp. 15859–15864.
15. Kolpashchikov D. et al. FABRIK-Based Inverse Kinematics For Multi-Section Continuum Robots. Proceedings of the 2018 18th International Conference on Mechatronics, 2018
16. GitHub - small23/Kuka_LBR_Toolbox. available at: https://github.com/small23/Kuka_LBR_Toolbox (accessed: 04.05.2022).

Лаптев Никита Витальевич, 1995г. рождения, аспирант ФГАОУ ВО НИ ТПУ, Инженер лаборатории НОЛ ОАБД ИШИТР, AuthorID: 1084295, SPIN-код: 2735-2600, ORCID: 0000-0003-0709-9974, nikitalaptev77@gmail.com.

Кравченко Андрей Александрович, 1998г. рождения, магистрант ФГАОУ ВО НИ ТПУ, AuthorID: 1091199, SPIN-код: 9114-4226, ORCID: 0000-0001-6828-3279, aak224@tpu.ru.

Гергет Ольга Михайловна, 1974г. рождения, д.т.н., профессор ФГАОУ ВО НИ ТПУ, профессор отделения информационных технологий, AuthorID: 115286, SPIN-код: 9352-4215, ORCID: 0000-0002-6242-9502, olgagerget@mail.ru.

UDC 681.571

DOI:10.38028/ESI.2022.26.2.015

Development of a library for planning and controlling the movements of the collaborative robot Kuka iiwa

Nikita V. Laptev, Andrey A. Kravchenko, Olga M. Gerget

National Research Tomsk Polytechnic University, Russia, Tomsk, nikitalaptev77@gmail.com

Abstract. The article presents a client-server library for the interaction of the Kuka LBR iiwa collaborative robot (on the KUKA Sunrise.OS operating system) with a remote PC. An intuitive high-level library implemented in the Matlab software package. Includes a server for the Kuka iiwa controller, as well as a client based on the MathWorks Matlab environment. The library's tool set includes more than 30 functions covering such operations as calculating forward and inverse kinematics, controlling the robot in Cartesian space and using sets of axis bends, path planning, graphical display and feedback. The developed software runs on a remote computer connected to the robot controller via the TCP/IP protocol. The article presents the developed methods and possible examples of robot control from the library.

Keywords: Collaborative robot, iiwa, KUKA Sunrise.OS, Matlab

Acknowledgement: The study has been funded with support from the Russian Foundation for Basic Research, projects no. 20-37-90055 and Russian Federation Governmental Program 'Nauka' no. FFSWW-2020-0014.

References

1. Hennemperger C. et al. Towards MRI-based autonomous robotic US acquisitions: a first feasibility study, IEEE Trans. Med. Imaging, 2016, vol. 36, no. 2, pp. 538–548.
2. Obal P., Gierlak P. EGM Toolbox—Interface for Controlling ABB Robots in Simulink. Sensors. Multidisciplinary Digital Publishing Institute, 2021, vol. 21, no. 22, pp. 7463.
3. Ostanin M., Yagfarov R., Klimchik A. Interactive Robots Control Using Mixed Reality. IFAC-PapersOnLine, 2019, vol. 52, no. 13, pp. 695–700.
4. LBR iiwa | KUKA in Malaysia. available at: <https://www.kuka.com/en-my/products/robotics-systems/industrial-robots/lbr-iiwa> (accessed: 19.04.2022).
5. KUKA Sunrise.OS | KUKA. available at: <https://www.kuka.com/en-us/products/robotics-systems/software/system-software/sunriseos> (accessed: 04.05.2022).
6. De Filippis L., Guglieri G., Quagliotti F. Path planning strategies for UAVS in 3D environments. Journal of Intelligent & Robotic Systems, Theory Appl, 2012, vol. 65, no. 1–4, pp. 247–264.
7. Hubicki C. et al. Walking and running with passive compliance: Lessons from engineering: A live demonstration of the atrias biped. IEEE Robotics & Automation Magazine, 2018, vol. 25, no. 3, pp. 23–39.
8. Chinello F. et al. Kuka control toolbox. IEEE Robotics & Automation Magazine, 2011, vol. 18, no. 4, pp. 69–79.

9. Sanfilippo F. et al. JOpenShowVar: an open-source cross-platform communication interface to kuka robots. 2014 IEEE International Conference on Information and Automation (ICIA). IEEE, 2014, pp. 1154–1159.
10. Sanfilippo F. et al. Controlling Kuka industrial robots: Flexible communication interface JOpenShowVar. IEEE Robotics & Automation Magazine, 2015, vol. 22, № 4. P. 96–109.
11. KUKA KR C4 | KUKA AG. available at: <https://www.kuka.com/en-us/products/robotics-systems/robot-controllers/kr-c4> (accessed: 04.05.2022).
12. Safeea M., Neto P. Kuka sunrise toolbox: Interfacing collaborative robots with matlab. IEEE Robotics & Automation Magazine, 2018, vol. 26, no. 1, pp. 91–96.
13. ROS: Home. available at: <https://www.ros.org/> (accessed: 04.05.2022).
14. Mokaram S. et al. A ROS-integrated API for the KUKA LBR iiwa collaborative robot. IFAC-PapersOnLine. Elsevier, 2017, vol. 50, no. 1, pp. 15859–15864.
15. Kolpashchikov D. et al. FABRIK-Based Inverse Kinematics For Multi-Section Continuum Robots. Proceedings of the 2018 18th International Conference on Mechatronics, 2018
16. GitHub - small23/Kuka_LBR_Toolbox. available at: https://github.com/small23/Kuka_LBR_Toolbox (accessed: 04.05.2022).

Nikita V. Laptev, born in 1995, postgraduate student, National Research Tomsk Polytechnic University, Engineer of Research Laboratory for Processing and Analysis of Big Data, AuthorID: 1084295, SPIN-code: 2735-2600, ORCID: 0000-0003-0709-9974, nikitalaptev77@gmail.com

Andrey A. Kravchenko, born in 1998, undergraduate student, National Research Tomsk Polytechnic University, Student of Division for Information Technology, AuthorID: 1091199, SPIN-code: 9114-4226, ORCID: 0000-0001-6828-3279, aak224@tpu.ru.

Olga M. Gerget, born in 1974, professor, National Research Tomsk Polytechnic University, Professor of Division for Information Technology, AuthorID: 115286, SPIN-code: 9352-4215, ORCID: 0000-0002-6242-9502, olgagerget@mail.ru.

Статья поступила в редакцию 08.05.2022; одобрена после рецензирования 06.06.2022; принята к публикации 10.06.2022.

The article was submitted 05/08/2022; approved after reviewing 06/06/2022; accepted for publication 06/10/2022.