

Ильичев Владимир Юрьевич, к.т.н., доцент кафедры «Тепловые двигатели и гидромашины», Калужский филиал ФГОУ ВО «Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет), г. Калуга, Россия

e-mail: patrol8@yandex.ru

Жуков Янислав Дмитриевич, студент кафедры «Мехатроника и робототехнические системы»

Калужский филиал ФГОУ ВО «Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет), г. Калуга, Россия

РАЗРАБОТКА ОСНОВЫ КОНСТРУКЦИИ БИОНИЧЕСКОГО ШЕСТИНОГО РОБОТА

Аннотация: Статья посвящена разработке методики разработки основы бионического шестиногого робота. Описаны сферы применения подобных роботов, их достоинства, отмечена актуальность их использования в современном мире. Указана цель работы, а также кратко описаны начальные этапы его конструирования. Особое внимание уделено созданию механических частей шестиногого робота, его схемы в системе Matlab из отдельных блоков (для проведения предварительных испытаний), после чего проведена отработка необходимой траектории его движения. Приведены исходные данные и результаты расчёта примера робота, созданного как в виде виртуальной, так и в виде механической конструкции. Описан следующий шаг работы над роботом, заключающийся в обоснованном выборе языка программирования Python для мини-компьютера Raspberry Pi и создания программы, реализующей все функции робота в окончательном варианте.

Ключевые слова: шестиногий робот, гексапод, программа Matlab,

траектория движения, компьютер Raspberry Pi, язык программирования Python.

Annotation: The article is devoted to the development of a methodology for developing the basis of a bionic six-legged robot. The areas of use of such robots, their advantages are described, the relevance of their use in the modern world is noted. The purpose of the work is indicated, as well as the initial stages of its design are briefly described. Particular attention is paid to the creation of mechanical parts of the six-legged robot, its scheme in the Matlab system from separate units (for preliminary tests), after which the necessary trajectory of its movement was developed. The initial data and results of calculating an example of a robot created both in the form of a virtual and in the form of a mechanical structure are given. The next step in working on the robot is described, which consists in the reasonable choice of the Python programming language for the Raspberry Pi mini-computer and the creation of a program that implements all the functions of the robot in the final version.

Keywords: six-legged robot, hexapod, Matlab program, motion path, Raspberry Pi computer, Python programming language.

Введение

Конструирование роботов началось много лет назад - первый чертёж робота, похожего на человека, был выполнен Леонардо да Винчи около 1495 года [1]. В настоящее время роботы предназначаются для выполнения множества целей и сферы их применения всё более расширяются [2; 3]:

- автоматизация технологических процессов производства;
- помощь человеку в повседневной жизни;
- проведение исследований в сложных или недоступных для человека условиях, например, в космосе;
- развлекательные и прочие цели.

Следует особенно отметить, что конструкция роботов часто заимствована из живой природы – например, внешне и по набору выполняемых функций они

вполне могут быть похожи на живых существ. Роботы работают согласно заложенной в них программе, поэтому правильное программирование в этом случае является особенно важным [4].

Большое распространение получили роботы на колёсах и ногах, обладающие особой мобильностью. Например, робота можно выполнить шестиногим, в виде паука, тогда он как правило имеет 6 степеней свободы – вращение и поступательное движение в 3 направлениях. Ноги крепятся к специальному приводному механизму, размещённому на основании или платформе робота. Такой робот обладает особой устойчивостью и может продолжать функционировать даже при отсутствии части конечностей.

Важно, что современные роботы, как правило, выполняют автономными – то есть они самостоятельно могут принимать решение о движении ног (для обеспечения устойчивости и необходимого направления движения) согласно показаниям, снимаемым с закреплёнными на них датчиках и подаваемым на систему управления, работающую под действием реализованной системы технического зрения [5].

Целью описываемой работы является выбор средств, а также описание непосредственно процесса разработки виртуальной модели шагающего робота, движение которого соответствует любой заданной траектории.

Материал и методы исследования

В настоящее время шагающий робот выполняется с использованием ставших стандартными методов его изготовления и программирования.

В описываемом в данной работе проекте используются следующие средства:

- корпус робота изготавливается из склеенных пластин из оргстекла;
- в корпус устанавливается специальная электронная плата расширения возможностей роботов – Robot Hat;
- программа робота «защита» в компьютер Raspberry Pi [6] и выполнена на самом популярном языке программирования Python [7];
- также в корпус помещается отсек для аккумуляторных батарей;

- двигателем ног робота является специальный сервопривод 15x MG90S;
- в качестве датчиков используются: видеочамера Raspberry Pi, гироскопическое устройство MPU6050, датчик ультразвуковых колебаний HC-SR04.

Чертежи для вырезания механических элементов робота были разработаны авторами в САD системе 3D моделирования SolidWorks [8]. К данным элементам относятся:

1. платформа для закрепления сервоприводов (рис. 1).

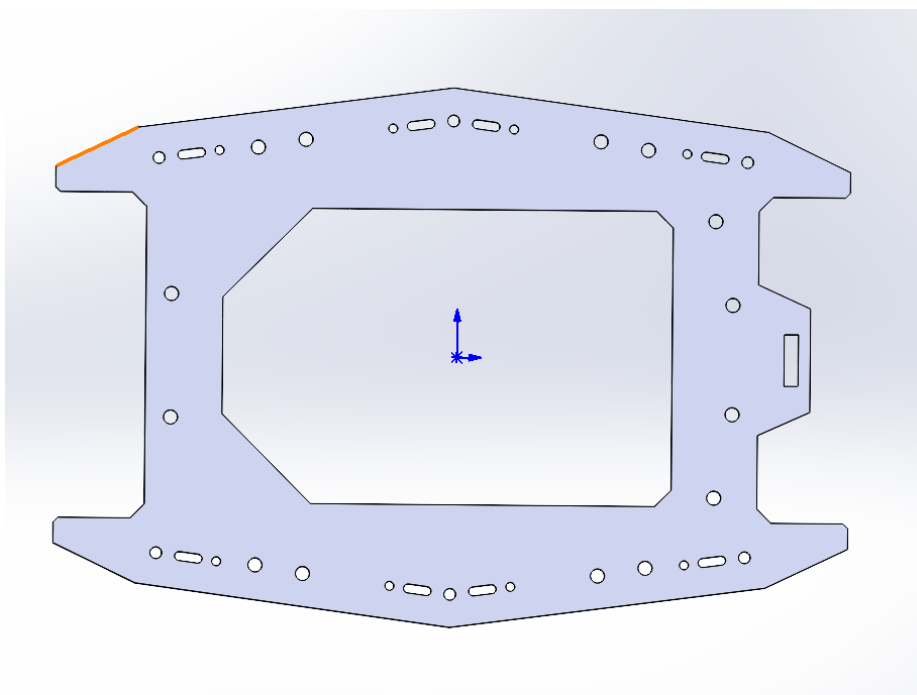


Рис. 1. Созданная в программе SolidWorks платформа для закрепления сервоприводов робота.

2. тазовая часть робота, созданная в виде сборки с самим сервоприводом 15x MG90S (рис. 2).

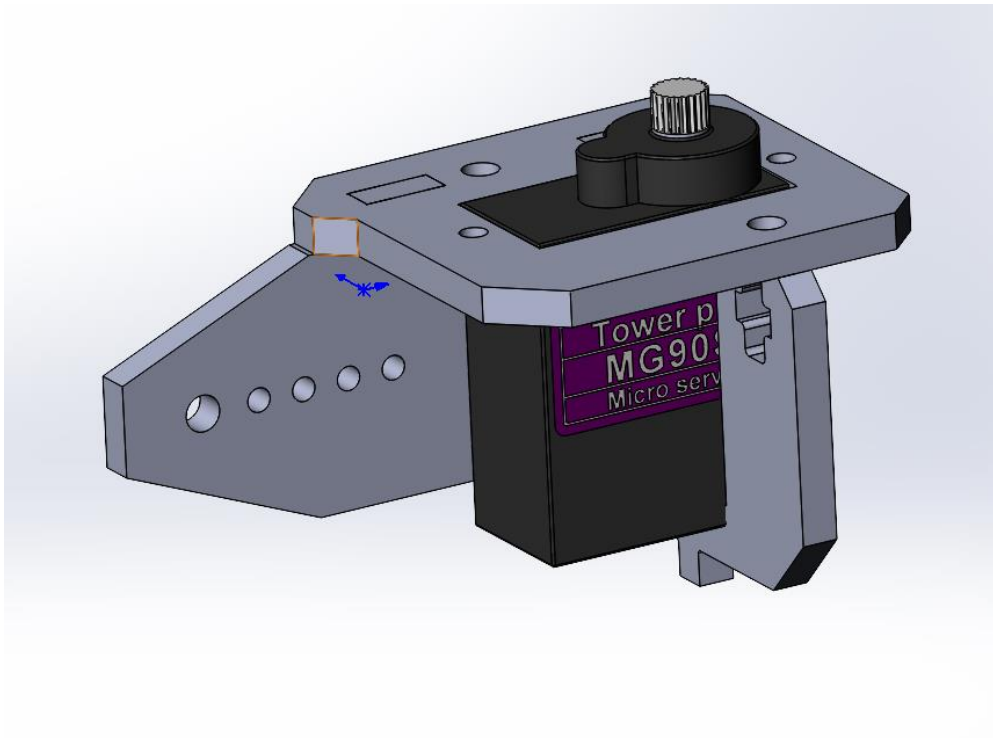


Рис. 2. Тазовая часть шагающего робота в сборке с заводским сервоприводом.

3. коленно-голеностопная часть ноги в количестве 6 экз. (по числу ног робота), выполненная как единое целое (рис. 3).

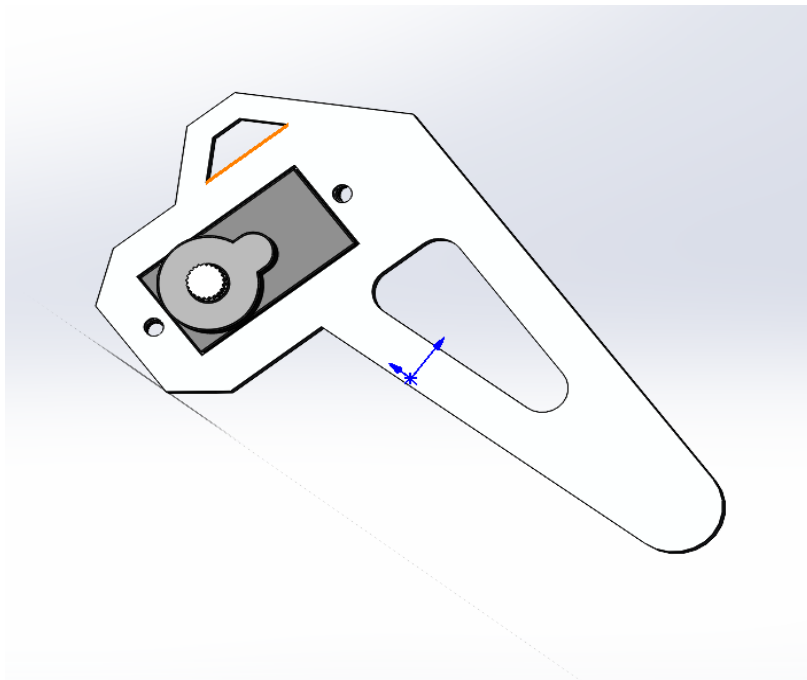


Рис. 3. Коленно-голеностопная часть ноги робота.

Соединив все спроектированные механические части шагающего робота (рис. 4), можно получить его упрощённую модель, движение которой уже можно исследовать с использованием виртуальных средств (в данной работе для этого была использована программа Matlab [9]).

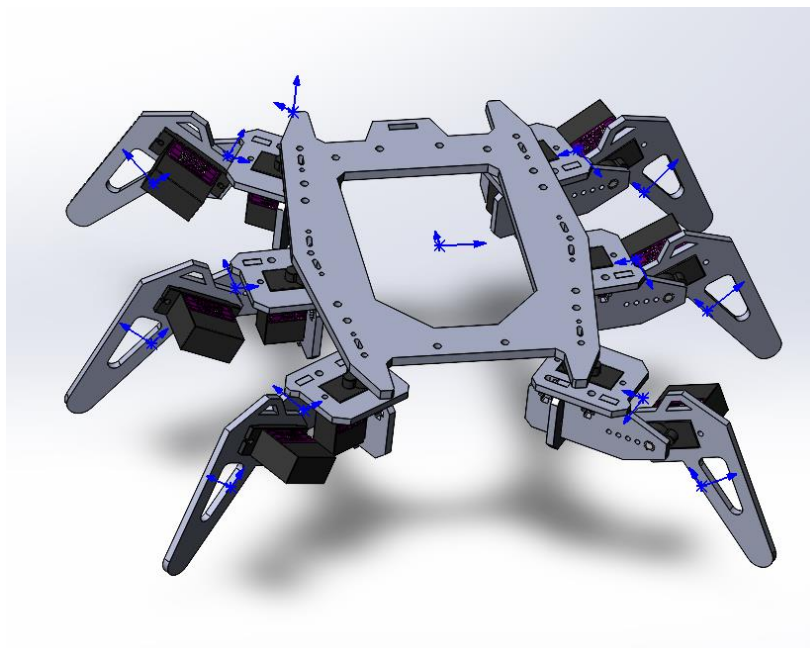


Рис. 4. Модель сборки механических частей шагающего робота.

Для исследования движения робота вначале необходимо экспортировать созданные в SolidWorks чертежи его деталей и сборок в соответствующий блок Matlab (Simmechanics) в формате STEP. Собранная конечность робота, например, в данном формате выглядит следующим образом (рис. 5) [10]:

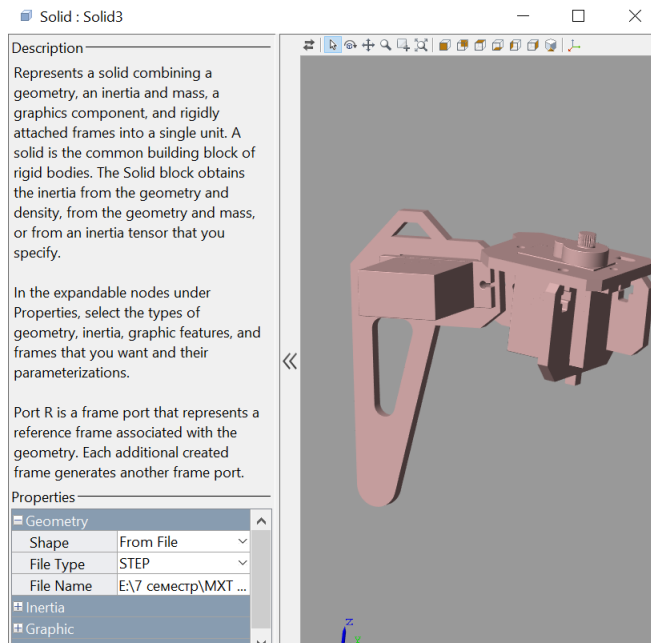


Рис. 5. Собранная конечность шагающего робота, импортированная в программу Matlab в формате STEP.

Каждый блок Matlab, кроме визуализации элемента робота, должен содержать описание возможностей его передвижения. Рассмотрим созданные элементы схемы робота Matlab (рис. 6) более подробно.

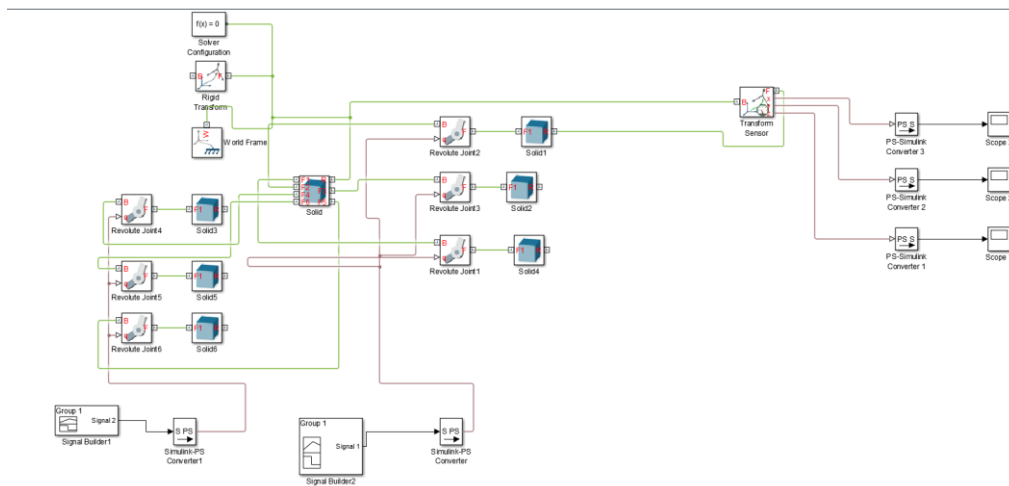


Рис. 6. Схема шагающего робота в программе Matlab.

Блок «Solver Configuration» (рис. 7) [11] описывает основные настройки решателя задачи движения робота. Без данного блока запуск проекта невозможен (так как никакая задача решаться не будет).

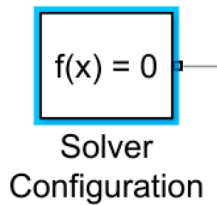


Рис. 7. Блок схемы робота «Solver Configuration» в программе Matlab.

Блок «World Frame» (рис. 8) задаёт свойства глобальной системы координат (даже при добавлении нескольких подобных блоков в проект Matlab глобальная система координат, относительно которой вычисляется перемещение элементов робота, не изменится).

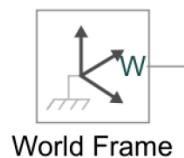


Рис. 8. Блок схемы робота «World frame» в программе Matlab.

Блок «Solid» (рис. 9) позволяет создать геометрию самостоятельно - в данном проекте он импортирует геометрию из собственноручно созданных файлов SolidWorks.

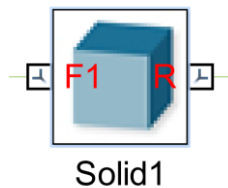


Рис. 9. Блок схемы робота «Solid» в программе Matlab.

Блок «Revolute Joint» (рис. 10) задаёт возможное вращательное движение конечностей проектируемого с помощью проекта шагающего робота.

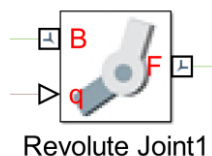


Рис. 10. Блок схемы робота «Revolute Joint» в программе Matlab.

Блок «Transform Sensor» (рис. 11) позволяет задавать необходимую задержку для выполнения программы, реализуемой схемой.

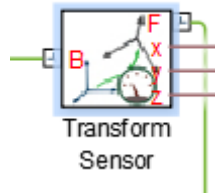


Рис. 11. Блок схемы робота «Transform Sensor» в программе Matlab.

Блок «Rigid Transform» (рис. 12) позволяет выполнять преобразование системы координат. Сначала осуществляет поворот осей, а после этого происходит перемещение исходной точки координатной системы.

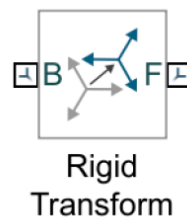


Рис. 12. Блок схемы робота «Rigid Transform» в программе Matlab.

Блок PS-Simulink Converter (рис. 13) – блок для преобразования физического сигнала (сигнала с любого входящего в схему датчика) в выходной сигнал Simulink.

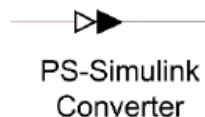


Рис. 13. Блок схемы робота «PS-Simulink Converter» в программе Matlab.

Блок Scope (осциллограф) (рис. 14) может отображать сигнал, генерируемый во время симуляции.



Рис. 14. Блок схемы робота «Score» в программе Matlab.

Блок «Signal Builder» (рис. 15) реализует возможность подачи сигнала на кинематическую цепь между платформой и конечностями робота.

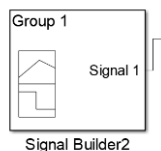


Рис. 15. Блок схемы робота «Signal Builder» в программе Matlab.

Описанный набор блоков Matlab позволяет подавать и снимать сигналы как с левой, так и с правой стороны спроектированного робота.

Пример расчёта

Для разработки программы отработки сигналов вначале в системе Matlab был произведён расчёт примера. Для этого были заданы траектории движения робота в пространстве, примеры которых представлены в примере (рис. 16).

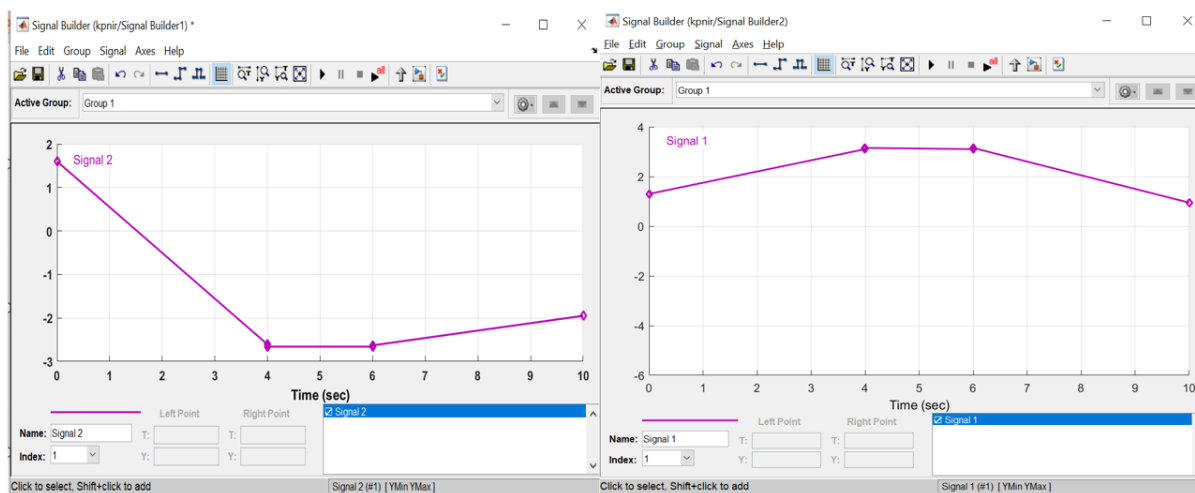


Рис. 16. Экспериментальные траектории движения робота в пространстве.

В результате многочисленных экспериментов удалось добиться уверенного отработки задаваемых траекторий на модели робота (рис. 17), а также устойчивого его положения при условиях, когда это возможно кинематически [12].

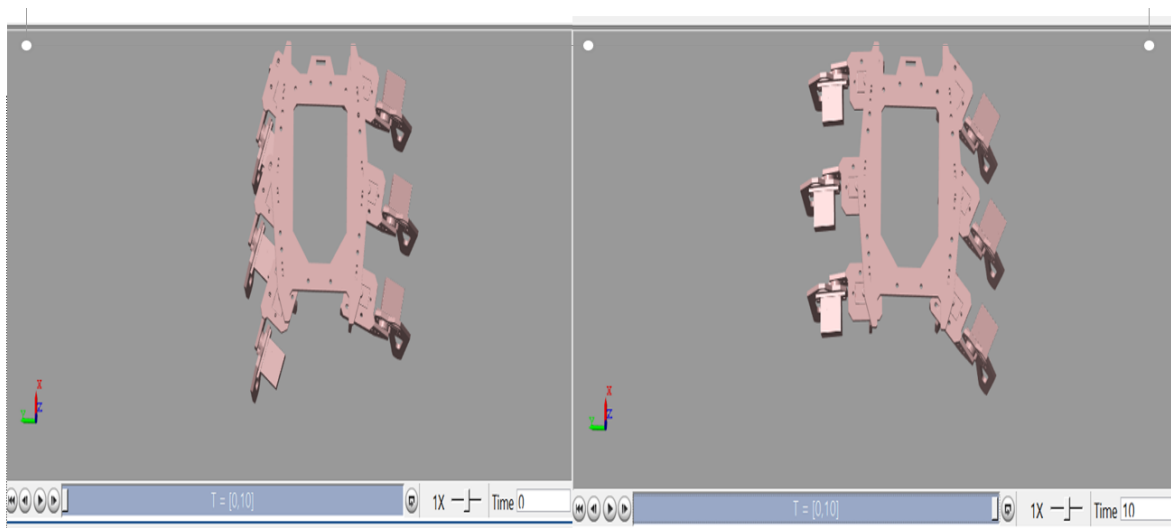


Рис. 17. Виртуальное движение робота в разных точках экспериментальных траекторий.

Заключение

Таким образом, цель работы достигнута: выбраны как виртуальные, так и материальные средства изготовления и испытания шагающего робота. Отработан алгоритм, позволяющий роботу уверенно двигаться по любой заданной траектории (пока только виртуально).

Следующим шагом описываемой работы будет являться программирование исследованного в данной работе движения робота с помощью применения описанных в статье аппаратных средств.

Для процессорно-аппаратной платформы робота Raspberry Pi доступно много языков программирования. Из них выбран самый популярный [13] на данный момент язык Python, для которого существует множество уже исследованных авторами библиотек функций программирования роботов, в том числе реализации компьютерного зрения, определения размера до препятствий с помощью ультразвука [14] и т.п.

Python является интерпретирующим, объектно-ориентированным и динамическим языком программирования высокого уровня с мощными функциями, хорошей совместимостью и высокой надежностью. Кроме того, он и библиотеки его функций являются свободно распространяемыми, т.е. бесплатными для использования. Большинство работ авторов, так же, как и описываемая, согласно перечисленным причинам, выполнены именно на языке

Python.

Библиографический список:

1. Булавинцева Т.П. Думать, догадываться, творить. // Начальная школа. 2016. № 11. С. 19-22.
2. Ильичев В.Ю. Организация управления мобильными объектами в робосимуляторе путем передачи команд из программы на языке Python по локальному IP. // Системный администратор. 2021. № 11 (228). С. 72-74.
3. Ильичев В.Ю. Создание скриптов Python для управления роботами в симуляторе программы FreeCAD. // Заметки ученого. 2021. № 11-1. С. 181-184.
4. Pichev V.Y. Development of program for determination of fractal dimensions of images. // International Research Journal. 2021. № 4-1 (106). С. 6-10.
5. Чухраев И.В., Ильичев В.Ю. Распознавание характерных объектов на изображении с использованием технологий компьютерного зрения. // E-Scio. 2021. № 8 (59). С. 122-131.
6. Фомин Д.С. Сравнительная характеристика операционных систем для Raspberry PI 3. // В сборнике: XLVII Огарёвские чтения. Материалы научной конференции. В 3-х частях. Составитель А.В. Столяров. Ответственный за выпуск П.В. Сенин. 2019. С. 389-391.
7. Ильичев В.Ю. Разработка программных продуктов с использованием модуля Python CoolProp для исследования эффективности утилизации тепла продуктов сгорания газообразных топлив. // Системный администратор. 2020. № 11 (216). С. 80-83.
8. Гузнецов В.Н., Журбенко П.А. SolidWorks при подготовке машиностроителя. // В сборнике: Будущее машиностроения России. Сборник докладов Двенадцатой Всероссийской конференции молодых ученых и специалистов (с международным участием). 2019. С. 891-894.
9. A new look for Matlab. // Machine Design. 2001. Т. 73. № 2. С. 28.
10. Донченко М.С. Моделирование руки манипулятора в среде Matlab Simulink. // В сборнике: Программная инженерия: методы и технологии

разработки информационно-вычислительных систем (ПИИВС-2020). сборник научных трудов III Международной научно-практической конференции (студенческая секция). Донецк, 2020. С. 217-220.

11. Комаров Е.Д., Руппель А.А. Математическое моделирование сложных технических систем с помощью Simulink. // Автоматика и программная инженерия. 2013. № 2 (4). С. 71-81.

12. Притыкин Ф.Н. Виртуальное моделирование движений роботов, имеющих различную структуру кинематических цепей. // Омск, 2014.

13. Ильичев В.Ю. Использование алгоритма дифференциальной эволюции для решения оптимизационных задач. // Системный администратор. 2021. № 4 (221). С. 80-83.

14. Любецкий Д.С., Фокин О.С. Обнаружение препятствий ультразвуком. // В книге: Актуальные проблемы физической и функциональной электроники. Материалы 23-й Всероссийской молодежной научной конференции. 2020. С. 158-159.