

## ПРОГРАММНЫЙ КОМПЛЕКС ДЛЯ ИССЛЕДОВАНИЯ НЕЙРОСЕТЕВЫХ ТЕХНОЛОГИЙ В СОСТАВЕ СИСТЕМЫ УПРАВЛЕНИЯ СФЕРИЧЕСКИМ ПРИВОДОМ ПРЯМОГО УПРАВЛЕНИЯ

А.В. Агеев

alekseu147147@yandex.ru

SPIN-код: 3341-0821

МГТУ им. Н.Э. Баумана, Москва, Российская Федерация

---

### Аннотация

Цель исследования — формирование программной среды для изучения и моделирования системы управления сферическим приводом прямого управления (СППУ) с использованием нейросетевой архитектуры. В качестве инструментального программного обеспечения рассмотрены системы моделирования MATLAB, Gazebo и нейросетевая библиотека Tensorflow. Особенность задачи моделирования данной системы управления заключается в том, что эта компьютерная модель должна иметь возможность интегрироваться в систему моделирования других частей автономного мобильного робота. Для решения проблемы интеграции применена система Robotic Operating System (ROS), позволяющая разрабатывать программные модули для подключения систем, которые изначально не поддерживают данную возможность.

### Ключевые слова

Нейронная сеть, автоматическое управление, компьютерное моделирование, мобильный робот, Robotic Operating System, MATLAB, Gazebo, Tensorflow

Поступила в редакцию 23.03.2020

© МГТУ им. Н.Э. Баумана, 2020

---

**Введение.** При проектировании автономных мобильных роботов (АМР) важным этапом является построение математической модели этой системы. Это связано с тем, что на качество функционирования системы в реальном мире влияет очень много факторов. При построении модели выделяют важные параметры среды и системы, которые затем оценивают. Как правило, создаваемые модели не имеют простого аналитического решения, поэтому применяется компьютерное моделирование, с помощью которого приближенно оцениваются параметры модели.

Цель работы — формирование программной среды для исследования и моделирования системы управления сферическим приводом прямого управления (СППУ) в составе системы технического зрения со структурированным подсветом, основанной на нейросетевых алгоритмах. Применение нейронной сети в качестве алгоритма управления обусловлено тем, что она имеет «предсказательную силу» (т. е. можно собрать данные о функционировании системы при различных начальных условиях и различных управляющих воздействиях и на данном массиве соответствий обучить нейронную сеть). Таким образом, нейросете-

вой алгоритм выступает в роли универсального аппроксиматора, который выявляет закономерности в законе управления. Это позволяет управлять системой без обратной связи, когда управление осуществляется в штатном режиме, в этом случае можно построить более качественный закон управления, так как устраняется задержка от системы обратной связи. Конечно, от этой системы нельзя избавиться, поскольку необходимо контролировать текущее состояние объекта управления, и если он выходит из штатного режима управления (т. е. возникает большая погрешность управления), необходимо строить управляющее воздействие уже с учетом обратной связи.

**Сферический привод прямого управления.** Система технического зрения со структурированным подсветом состоит из видеокамеры, управляющего устройства (в данном случае это персональный компьютер) и СППУ, выполняющей подсветку области интереса. Сферический привод [1] представляет собой карданов подвес, в котором закреплен лазерный целеуказатель и постоянный магнит. Карданов подвес окружают две пары соосных соленоидов. Одна пара расположена по вертикали, а другая — по горизонтали. При подаче напряжения на катушки возникает магнитное поле, на которое реагирует постоянный магнит. С помощью микроконтроллера АТМega128 осуществляется управление напряжением на катушках. Меняя это напряжение, можно вращать закрепленную в подвесе систему по определенной траектории. Внутри системы установлены четыре датчика Холла, по показаниям которых можно определить текущее состояние объекта управления.

Назначение СППУ — подсветка области интереса, которую снимает видеокамера. Полученное изображение поступает в управляющее устройство, которое обрабатывает такие изображения, на основе полученных данных формирует управление роботом и выдает новое задание для микроконтроллера СППУ. Таким образом, существует иерархическая система управления:

- 1) микроконтроллер СППУ, выполняющий вращение лазерным целеуказателем;
- 2) управляющее устройство, которое выдает задания для СППУ и выполняет управление роботом.

Для исследования нейросетевого подхода в задаче управления СППУ необходимо сформировать программную среду, позволяющую проводить эксперименты как с компьютерными моделями СППУ и АМР, так и с реальными образцами устройств.

**Программное обеспечение для моделирования.** Для выполнения моделирования физических систем существует множество различных программных решений: от физических движков, которые предоставляют возможность расчета сил, действующих на моделируемые объекты, до систем моделирования, которые предоставляют не только алгоритмы расчета сил, но и визуализацию 3D-мира, в котором действует объект, и иные инструменты для построения среды, в которой моделируется система.

В работе [2] рассмотрены физические движки, такие как NVidia PhysX, ODE, Bullet, разработанный для моделирования физики игрового мира, XDE, OpenHRP, предназначенные для моделирования роботов и иных технических средств, системы моделирования Gazebo, V-Rep, Webots, Unity и др. Такое разнообразие программных решений обусловлено тем, что при моделировании накладываются различные ограничения и в данных системах используются разные схемы моделирования. Например, PhysX был разработан с целью моделирования игрового мира, таким образом, на первом месте для этой системы стоит производительность, а не точность вычислений. Этот физический движок используется в системе Unity, являющейся игровым движком, применяемым для компьютерного моделирования [3].

При разработке систем важна визуализация данных, поскольку так их легче интерпретировать. Для этой цели было разработано множество различных инструментов [4, 5], такие как QReal:Robots, Microsoft Robotics Developer Studio, предназначенные для проектирования робототехнических систем.

При проектировании важна организация вычислений. Для этого была разработана система Robot Operating System, предназначенная для объединения различных программных продуктов в единую систему. Разработчики робота OpenRoACH [6], который представляет собой шестиного шагающего робота, выложили систему в открытый доступ, что позволяет оценить возможности ROS и совместно используемого программного обеспечения.

**Структура программного комплекса.** Для моделирования СППУ и построения его системы управления, основанной на нейросетевой архитектуре, предлагается использовать следующее программное обеспечение: MATLAB Simulink, Robotic Operating System (ROS), Gazebo, Tensorflow, на основе которых можно составить три программные структуры, изображенные на рисунке. Программный комплекс должен поддерживать следующие возможности:

- 1) интеграция с реальным устройством (связь с микроконтроллером);
- 2) интеграция разработанной модели в другие системы моделирования.

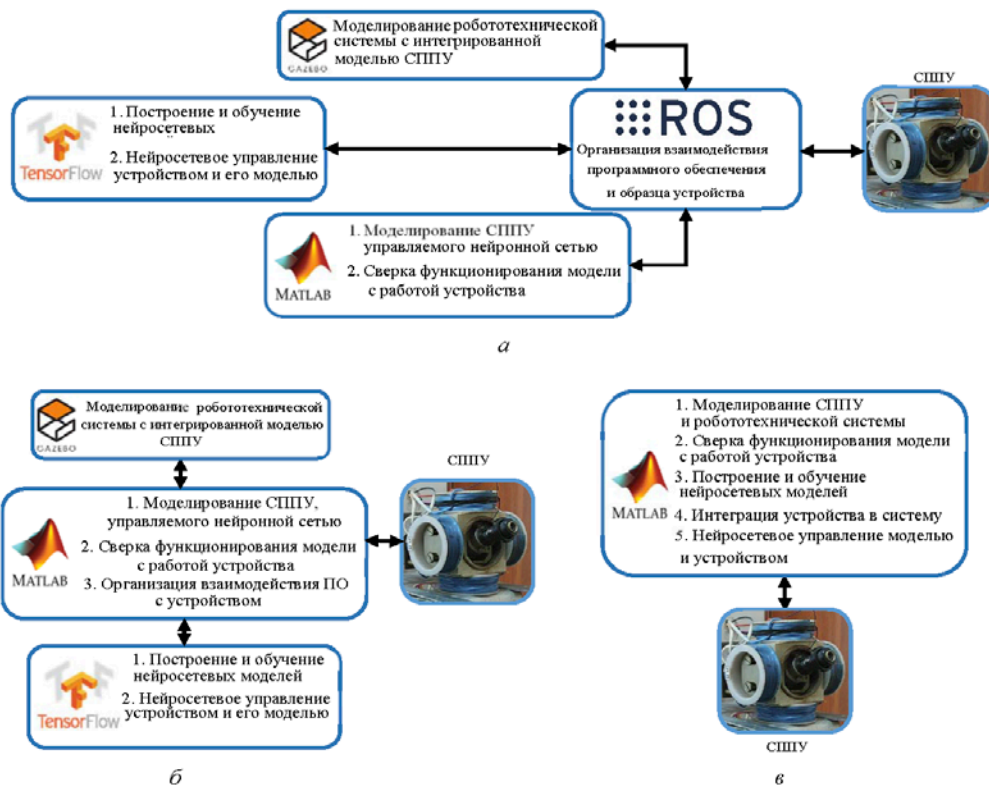
Возможность интеграции программной системы с реальным устройством позволяет оперативно организовать применение разработанного алгоритма управления на устройстве. Таким образом, в процессе разработки становится возможным сопоставить компьютерную модель и устройство, минуя этап отдельной разработки программы для микроконтроллера.

Необходимость второго пункта заключается в том, что СППУ является частью АМР, поэтому после того как будет разработана модель СППУ (компьютерная модель устройства, выполняющая моделирование физических процессов и алгоритмы управления), ее необходимо будет интегрировать в модель АМР. При этом модель АМР может быть разработана в другой системе моделирования.

Рассмотрим первую схему (рис. а). В данном случае система ROS выступает в роли связующего звена, которое выполняет интеграцию системы моделирования Gazebo с системой MATLAB (в ней реализована модель СППУ), Tensorflow (моделирование нейронных сетей) и ко всей этой системе подключено реальный образец СППУ. Данная схема обладает следующими достоинствами:

1) ROS является системой, объединяющей широко используемые библиотеки (например, OpenCV) в единую инфраструктуру. Предоставляет инструменты для проектирования распределенной системы управления робототехническими комплексами. У программной системы имеется активное сообщество разработчиков. Таким образом, можно найти готовые программные модули, позволяющие выполнить интеграцию внешних систем с ROS. Например, системы Gazebo, MATLAB имеют готовые расширения, благодаря которым можно подключить эти системы к ROS;

2) микроконтроллер подключен ко всей системе, что позволяет сравнивать модельные показатели с реальными показателями поведения СППУ.



Схемы программного комплекса для исследования и моделирования нейросетевого алгоритма управления СППУ

В принципе, можно отказаться от ROS и использовать MATLAB в качестве связующего звена (рис. б), поскольку MATLAB поддерживает интеграцию с AVR-контроллером, Gazebo и Python. Но тут возникают следующие проблемы:

1) ограничение на количество программных продуктов, которые могут быть интегрированы в систему (было бы полезно, ощутив недостаток функциональности, оперативно подключить новый программный продукт);

2) возможности интеграции MATLAB с аппаратной платформой ограничено Arduino и списком ПЛИС от Intel, что является существенным ограничением;

3) MATLAB, например, поддерживает только определенные версии языка Python. В случае обновления MATLAB придется устанавливать другие поддерживаемые версии интегрируемых продуктов.

Вторую проблему можно решить с использованием расширения Embedded Coder, но оно не позволяет взаимодействовать с устройством в реальном времени, поскольку выполняет трансляцию модели Simulink на язык C/C++. После трансляции программы ее необходимо собрать с помощью компилятора целевой платформы и через программатор загрузить в микроконтроллер.

Третья возможная схема (рис. в) — использование только системы MATLAB со следующими установленными в нее расширениями:

- Simscape (моделирование физических систем);
- Virtual Reality Toolbox (3D-визуализация);
- Neural Network Toolbox (построение и обучение нейронных сетей);
- Embedded Coder (пакеты для трансляции моделей Simulink в на язык C/C++):
  - MATLAB Coder;
  - Simulink Coder;
  - Fixed-Point Designer;
- компилятор целевой платформы (например, Atmel Studio).

Недостатки третьей схемы такие же, как и у второй (см. рис. б), но при этом все построения ограничены только возможностями системы MATLAB. Поэтому в качестве структуры комплекса для исследования предлагается использовать первую схему (см. рис. а).

**Симулятор Gazebo.** В качестве системы 3D-моделирования был выбран симулятор Gazebo. Это бесплатная программа с открытым исходным кодом и качественной документацией, которая поддерживает интеграцию с ROS. Поскольку система дает возможность писать модули на языке C++, что позволяет организовать такой же интерфейс управления моделью, как на устройстве [7], становятся возможными отладка алгоритмов управления на модели робота, а затем оперативное тестирование на реальном образце устройства.

Для расчета физических процессов моделируемой системы симулятор использует внешний физический движок Open Dynamics Engine, который разработан для моделирования кинематики и динамики твердого тела. Для визуализации используется система OpenGL/GLUT, которая имеет низкое качество графики. Это сделано специально, чтобы сосредоточить все ресурсы на моделировании, но если необходимо улучшить изображение (например, для презентации), можно подключить графический движок OGRE.

Симулятор Gazebo создает прослойку между физическим движком и пользователем, предоставляет собственный API (application programming interface — программный интерфейс приложения). При необходимости можно установить другой физический движок. Система поддерживает следующие системы расчета физических процессов:

- Open Dynamics Engine (ODE);
- Bullet;

- Dynamic Animation and Robotics Toolkit (DART);
- Simbody.

Возможность поменять физический движок является полезной. Каждая система была разработана своим собственным сообществом, мотивированным определенной областью применения, от игровой (Bullet) до упрощенной динамики робота (ODE), от биомеханики (Simbody) до компьютерной графики и управления роботом (DART).

Описание моделируемой системы в Gazebo осуществляется с помощью языка XML, в котором задаются как физические параметры системы, так и визуальные параметры. Существуют также модели роботов, уже готовые для использования. Это является плюсом при изучении данной системы.

При моделировании предоставляются следующие типы объектов:

- 1) модели — любые объекты, которые включают в себя физическое и геометрическое представления. Модель состоит из твердых тел, соединенных суставами;
- 2) тела — это примитивы, описывающие геометрическую составляющую объекта и имеющие физические параметры;
- 3) суставы — соединения двух и более тел для организации кинематических и динамических связей между этими телами;
- 4) датчики — объекты, оперирующие информацией в системе. Существуют статические датчики, собирающие информацию, и динамические датчики, излучающие информацию в систему;
- 5) интерфейсы — данные объекты предназначены для построения API для взаимодействия внешних программ с моделируемой системой;
- 6) статические объекты — объекты, предназначенные для имитации неподвижного окружения.

Симулятор Gazebo также предоставляет возможность оперативного построения окружающей среды. Например, можно построить для тестирования робота лабиринт, изобразив его схему в svg-формате, по контуру лабиринта будут сформированы стены.

Моделирование физических систем является трудным для вычисления процессом, особенно когда объект моделирования сложный или их много, поэтому для проведения моделирования таких систем нужно мощное оборудование. Для решения этой проблемы Gazebo предоставляет клиент-серверную архитектуру (толстый сервер, тонкий клиент), в которой весь расчет физических процессов можно организовать на отдельной системе (компьютерная система в дата-центре), а весь графический интерфейс и инструментарий оставить на клиентской машине. Для увеличения производительности моделирования хотелось бы иметь возможность выполнять распределенные вычисления, но поскольку физические движки имеют монолитную архитектуру, это невозможно. Разработчики Gazebo применяют различные ухищрения для использования всех ресурсов компьютера во время моделирования. Например, они определяют не взаимодействующие области моделируемого пространства в данный момент време-

ни имитации, и затем выполняют расчет физических процессов для этих областей параллельно [8].

**Нейросетевая библиотека Tensorflow.** Tensorflow является нейросетевой библиотекой, которую можно использовать в языке Python. Она предоставляет два интерфейса: высокоуровневый — позволяющий выполнять быстрые построения нейросетевых моделей (библиотека Keras), и низкоуровневый, предоставляющий обширные возможности по построению собственных нейросетевых моделей. Этот интерфейс позволяет реализовать базовые операции над тензорами, функции активации, методы регуляризации, вычисление свертки, производных и много другое. Во время оперирования библиотекой вызываемые математические операции не выполняются, вместо этого строится граф вычислений, а затем с помощью объекта `session` запускается процесс вычислений. Библиотека поддерживает как AVX-ускорения на центральном процессоре компьютера, так и вычисления на видеокартах от компании NVidia с помощью технологии CUDA. Библиотека обладает следующими преимуществами: 1) имеет подробную документацию; 2) в развитии библиотеки участвует большое сообщество, что позволяет быстро начать работу и пользоваться этим продуктом от Google. Поскольку библиотека предоставляет низкоуровневый интерфейс, с помощью нее можно относительно быстро реализовывать самые последние алгоритмы, представленные в статьях, которые еще не были добавлены в высокоуровневую часть интерфейса разработчиками библиотеки.

Особенность Tensorflow заключается в том, что она имеет оптимизирующее компиляторы для разных платформ [9], таким образом, построенную модель можно загрузить в микроконтроллер. Однако версия библиотеки, которая будет работать на контроллере, имеет ограничения:

- 1) поддерживается только определенный список устройств;
- 2) библиотека ограничена в функциональности по сравнению с полной версией (но для отсутствующих операций можно написать собственную реализацию, соблюдая интерфейс библиотеки).

**Система моделирования MATLAB Simulink.** MATLAB — это пакет прикладных программ, используемый в инженерных расчетах, предоставляющий среду моделирования Simulink. Данный пакет имеет систему плагинов, с помощью которой можно расширить области применения этого инструментария. Simulink — это среда имитационного моделирования, в которой с помощью блоков и направленных связей между ними строится имитационная модель. В базовой комплектации MATLAB Simulink предоставляет блоки для описания математических систем (например, блоки для построения дифференциальных и обыкновенных уравнений). Для решения описанных систем доступна возможность выбора алгоритма для проведения численного решения (например, метод трапеций, метод Дормана Принца 4–5-го порядка).

Расширения могут предоставлять блоки для математического моделирования, например, `toolbox simscape`. Данное расширение для Simulink предоставляет блоки для моделирования физических процессов в системе:

- моделирование механики;
- моделирование электрических компонентов;
- моделирование гидравлики;
- моделирование тепловых процессов.

Также отдельно поставляются дополнительные расширения для перечисленных выше компонентов.

Выполнив моделирование автономного мобильного робота с помощью Simscare, его можно поместить в некоторую окружающую среду, воспользовавшись расширением Virtual Reality Toolbox, которое предназначено для 3D-визуализации динамики устройства [10]. Данное расширение представляет собой программу для построения 3D-сцены и библиотеки для MATLAB, позволяющее соединить 3D-сцену с моделируемой системой. Но это расширение предоставляет только визуализацию, моделирование окружающего мира выполняется посредством других моделей Simulink.

Для моделирования функционирования робота удобнее воспользоваться расширением Robotics System Toolkit, которое предоставляет множество алгоритмов для проектирования и моделирования АМР, организует интеграцию с ROS, Gazebo. Все это превращает построение системы не в проектирование алгоритмов, а в создание системы управления на основе готовых блоков, предоставляющих те или иные алгоритмы и инструменты. При этом повышается производительность, такой подход меньше подвержен ошибкам. Все компоненты, используемые в среде Simulink, без всяких проблем могут взаимодействовать друг с другом, что позволяет быстро проектировать сложные системы, используя всевозможные расширения.

**Framework Robotics Operating System.** Система Robot operating system (ROS) представляет собой систему для организации распределенного взаимодействия между программами. Программа предоставляет концепцию «издатель — подписчик», при которой одна программа отправляет сообщения в ROS, а другая получает эти сообщения и выполняет собственные вычисления на основе этих данных. В данную систему интегрируются такие системы, как MATLAB, Gazebo, OpenCV (библиотека машинного зрения), MoveIt (система расчета траектории подвижных частей робота), и множество других библиотек, например, библиотеки для построения карт препятствий [11].

Благодаря используемой концепции можно проектировать робота, используя «тяжелые» продвинутые инструменты, а после того как система будет разработана, инструментарий, интегрированный в ROS, можно заменить собственными легковесными (менее требовательными к аппаратному обеспечению) реализациями. Программы, работающие в системе ROS, называют узлами. Существует главный процесс (master), к которому подключаются узлы, тем самым обнаруживая друг друга, и начинают взаимодействовать. Достоинство данной архитектуры в том, что неважно, где находятся узлы, самое главное, чтобы сохранялась их связь с мастер-процессом. В работе [12] при реализации робота, движущегося за объектом, производительности одноплатного компьютера для



обработки изображений с видеокамеры не хватало, поэтому узел с библиотекой OpenCV с устройства перенесли на стационарный компьютер. Робот стал работать лучше, и переписывать программное обеспечение автору не пришлось.

**Выводы.** Для исследования нейросетевых алгоритмов в задаче управления сферическим приводом, являющимся частью автономной робототехнической системы, было предложена структура программного комплекса, которая обеспечивает следующие возможности:

1) построение модели единой робототехнической системы из частей, моделируемых в разных системах;

2) подключение реального устройства к программному комплексу для получения возможности управления как моделью, так и устройством, что позволяет выполнить верификацию модели и настройку алгоритмов управления.

Данные возможности реализуются благодаря применению *Robotic operating system*, которая и выполняет объединение различных программных продуктов. Используемое программное обеспечение изначально поддерживает интеграцию с данной системой, но если используемая программа не поддерживает ROS, можно написать программу-адаптер на языке Python.

ROS является бесплатным продуктом с открытым исходным кодом и большим развивающимся сообществом. Существует поддержка со стороны MATLAB, что позволяет быстро создавать макеты проектируемой системы для тестирования и анализа.

## Литература

- [1] Соколов С.М., Трифонов О.В., Ярошевский В.С. Система управления сферическим приводом прямого управления с датчиками Холла в контуре обратной связи. *Препринты ИПМ им. М.В. Келдыша*, 2008, № 94.
- [2] Клавдиенко А.А., Манюхина А.Н. Обзор инструментов моделирования бипедальных роботов. *Наука и Мир*, 2016, № 4(32), с. 52–58.
- [3] Гусев А.А., Молотов Р.С. Реализация взаимодействия имитационных моделей боевых машин в составе симулятора военного полигона в среде Unity. *Вестник УлГТУ*, 2016, № 3(75), с. 44–47.
- [4] Меженин А.В. Виртуальные 3D среды, как средство верификации и тестирования при проектировании. *Приоритетные научные направления: от теории к практике*, 2016, № 21, с. 105–110.
- [5] Михайлова У.В., Михайлов Е.А., Сарваров А.С. Программные решения для разработки архитектуры системы управления роботом. *Электротехнические системы и комплексы*, 2013, № 21, с. 111–117.
- [6] Wang L., Yang Y., Correa G., et al. Fearing OpenRoACH: a durable open-source hexapedal platform with onboard robot operating system (ROS). *ICRA*, 2019, pp. 9466–9472. DOI: <https://doi.org/10.1109/ICRA.2019.8794042>
- [7] Koenig N., Howard A. Design and use paradigms for gazebo, an open-source multi-robot simulator. *IEEE/RSJ IROS*, pp. 2149–2154. DOI: <https://doi.org/10.1109/IROS.2004.1389727>
- [8] Gazebo parallel physics report. DARPA robotics challenge. OSRF, 2015.

- 
- [9] MLIR: A new intermediate representation and compiler framework. *medium.com: веб-сайт*. URL: <https://medium.com/tensorflow/mlir-a-new-intermediate-representation-and-compiler-framework-beba999ed18d> (дата обращения: 10.11.2019).
- [10] Скобелев М.М. Разработка в Matlab-Simulink модели визуализации в виртуальной реальности управляемого пространственного движения подводного аппарата. *Наука и образование: научное издание*, 2011, № 10. URL: <http://technomag.edu.ru/doc/234292.html>
- [11] ROS. Стек навигации. *habr.com: веб-сайт*. URL: <https://habr.com/ru/post/327888/> (дата обращения: 14.12.2019).
- [12] Робот-тележка на ROS. Часть 3. Ускоряемся, меняем камеру, исправляем походку. *habr.com: веб-сайт*. URL: <https://habr.com/ru/post/463147/> (дата обращения: 14.12.2019).

**Агеев Алексей Владимирович** — студент кафедры «Программное обеспечение ЭВМ и информационные технологии», МГТУ им. Н.Э. Баумана, Москва, Российская Федерация.

**Научный руководитель** — Рудаков Игорь Владимирович, кандидат технических наук, доцент кафедры «Программное обеспечение ЭВМ и информационные технологии», МГТУ им. Н.Э. Баумана, Москва, Российская Федерация.

**Ссылку на эту статью просим оформлять следующим образом:**

Агеев А.В. Программный комплекс для исследования нейросетевых технологий в составе системы управления сферическим приводом прямого управления. *Политехнический молодежный журнал*, 2020, № 04(45). <http://dx.doi.org/10.18698/2541-8009-2020-04-596>

## SOFTWARE PACKAGE FOR THE STUDY OF NEURAL NETWORK TECHNOLOGIES AS PART OF A DIRECT SPHERICAL DRIVE CONTROL SYSTEM

A.V. Ageev

alekseu147147@yandex.ru

SPIN-code: 3341-0821

Bauman Moscow State Technical University, Moscow, Russian Federation

---

### Abstract

The purpose of the study is the formation of a software environment for studying and modeling a control system for a direct spherical drive (DSD) using a neural network architecture. Modeling systems MATLAB, Gazebo and the neural network library Tensorflow are considered as tool software. A feature of the control system modeling is that this computer model should be able to integrate into the modeling systems of other parts of an autonomous mobile robot. To solve the integration problem, the Robotic Operating System (ROS) was used, which allows developing software modules for connecting systems that initially do not support this feature.

### Keywords

Neural network, automatic control, computer simulation, mobile robot, Robotic Operating System, MATLAB, Gazebo, Tensorflow

Received 23.03.2020

© Bauman Moscow State Technical University, 2020

---

### References

- [1] Sokolov S.M., Trifonov O.V., Yaroshevskiy V.S. The control system of spherical direct drive actuator with Hall sensors in a feedback. *Preprinty IPM im. M.V. Keldysha* [KIAM Preprint], 2008, no. 94 (in Russ.).
- [2] Klavdienko A.A., Manyukhina A.N. The review of simulation tools for bipedal robots. *Nauka i Mir* [Science and world], 2016, no. 4(32), pp. 52–58 (in Russ.).
- [3] Gusev A.A., Molotov R.S. Implementation of interaction of combat vehicles models in the military simulator in Unity ide. *Vestnik UIGTU*, 2016, no. 3(75), pp. 44–47 (in Russ.).
- [4] Mezhenin A.V. Virtual 3D medium as verification and test tool at designing. *Prioritetnye nauchnye napravleniya: ot teorii k praktike*, 2016, no. 21, pp. 105–110.
- [5] Mikhaylova U.V., Mikhaylov E.A., Sarvarov A.S. Software solutions for the development of robot control system architecture. *Elektrotekhnicheskie sistemy i komplekсы* [Electro-technical Systems and Complexes], 2013, no. 21, pp. 111–117 (in Russ.).
- [6] Wang L., Yang Y., Correa G., et al. Fearing OpenRoACH: a durable open-source hexapedal platform with onboard robot operating system (ROS). *ICRA*, 2019, pp. 9466–9472. DOI: <https://doi.org/10.1109/ICRA.2019.8794042>
- [7] Koenig N., Howard A. Design and use paradigms for gazebo, an open-source multi-robot simulator. *IEEE/RSJ IROS*, 2004, pp. 2149–2154. DOI: <https://doi.org/10.1109/IROS.2004.1389727>
- [8] Gazebo parallel physics report. DARPA robotics challenge. OSRF, 2015.
- [9] MLIR: A new intermediate representation and compiler framework. *medium.com*: website. URL: <https://medium.com/tensorflow/mlir-a-new-intermediate-representation-and-compiler-framework-beba999ed18d> (accessed: 10.11.2019).

- [10] Skobelev M.M. Development of MATLAB-Simulink model of visualization in virtual reality of controlled spatial motion of underwater vehicle. *Nauka i obrazovanie: nauchnoe izdanie* [Science and Education: Scientific Publication], 2011, no. 10. URL: <http://technomag.edu.ru/doc/234292.html> (in Russ.).
- [11] ROS. Stek navigatsii [ROS navigation stack]. *habr.com: website* (in Russ.). URL: <https://habr.com/ru/post/327888/> (accessed: 14.12.2019).
- [12] Robot-telezka na ROS. Chast' 3. Uskoryaemsa, menyaem kameru, ispravlyаем pokhodku [ROS trolley robot, Part 3. Accelerating, changing camera, improving motion]. *habr.com: website* (in Russ.). URL: <https://habr.com/ru/post/463147/> (accessed: 14.12.2019).

**Ageev A.V.** — Student, Department of Computer Software and Information Technologies, Bauman Moscow State Technical University, Moscow, Russian Federation.

**Scientific advisor** — Rudakov I.V., Cand. Sc. (Eng.), Assoc. Professor, Department of Computer Software and Information Technologies, Bauman Moscow State Technical University, Moscow, Russian Federation.

**Please cite this article in English as:**

Ageev A.V. Software package for the study of neural network technologies as part of a direct spherical drive control system. *Politekhnicheskiiy molodezhnyy zhurnal* [Politechnical student journal], 2020, no. 04(45). <http://dx.doi.org/10.18698/2541-8009-2020-04-596.html> (in Russ.).