

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования

«Волгоградский государственный технический университет»

Факультет электроники и вычислительной техники

Кафедра «Электронно-вычислительные машины и системы»

Утверждаю

Зав. кафедрой

А. Е. Андреев  
(подпись) (инициалы, фамилия)

"\_\_" 20\_\_ г.

## ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к \_\_\_\_\_ выпускной работе бакалавра \_\_\_\_\_ на тему  
(наименование вида работы)

Разработка программно-аппаратного обеспечения пульта для управления  
инсектоморфным роботом

Автор \_\_\_\_\_ Кухно Дмитрий Андреевич  
(подпись и дата подписания) (фамилия, имя, отчество)

Обозначение ВРБ-40461806-10.44.05-20ПЗ  
(код документа)

Группа ИВТ-460  
(шифр группы)

Направление 09.03.01 Информатика и вычислительная техника  
(код по ОКСО, наименование направления, программы)

Профиль "Вычислительные машины, комплексы, системы и сети"

Руководитель работы \_\_\_\_\_ А.С. Горобцов  
(подпись и дата подписания) (инициалы и фамилия)

Консультанты по разделам:

(краткое наименование раздела) \_\_\_\_\_ (подпись и дата подписания) \_\_\_\_\_ (инициалы и фамилия)

Нормоконтролер \_\_\_\_\_ (подпись и дата подписания) \_\_\_\_\_ (инициалы и фамилия)

Волгоград, 2020 г.

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования

«Волгоградский государственный технический университет»

Факультет электроники и вычислительной техники

Кафедра «Электронно-вычислительные машины и системы»

Утверждаю

Зав. кафедрой

А. Е. Андреев  
(подпись) (инициалы, фамилия)

"\_\_" 20\_\_ г.

Задание на выпускную работу бакалавра

(наименование вида работы)

Студент Кухно Дмитрий Андреевич

(фамилия, имя, отчество)

Код кафедры 10.44 Группа ИВТ-460

Тема Разработка программно-аппаратного обеспечения пульта для  
управления инсектоморфным роботом

Утверждена приказом по университету от "20" марта 2020 г. № 377-ст

Срок представления работы \_\_\_\_\_  
(дата, подпись студента)

Исходные данные для выполнения работы

Документация к компонентам, модель робота-гексапода в системе ФРУНД

Содержание основной части пояснительной записи

## ВВЕДЕНИЕ

1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ СИСТЕМЫ УПРАВЛЕНИЯ ШАГАЮЩИМИ РОБОТАМИ
2. ПРОЕКТИРОВАНИЕ СИСТЕМЫ УПРАВЛЕНИЯ РОБОТОМ-ГЕКСАПОДОМ
3. РЕАЛИЗАЦИЯ СИСТЕМЫ УПРАВЛЕНИЯ РОБОТОМ ГЕКСАПОДОМ

4. ТЕСТИРОВАНИЕ СИСТЕМЫ УПРАВЛЕНИЯ РОБОТОМ-  
ГЕКСАПОДОМ  
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

## Перечень графического материала

1. Актуальность работы
2. Задачи
3. Аналоги
4. Архитектура системы управления
5. Реализация веб интерфейса управления
6. Тестирование управления в GAZEBO
7. Выводы

Руководитель работы \_\_\_\_\_  
(подпись и дата подписания)

А.С. Горобцов  
(инициалы и фамилия)

## АННОТАЦИЯ

Предлагается создание программного-аппаратного обеспечения пульта для управления инсектоморфным роботом. Реализованы подсистема обмена данными с программой "ФРУНД", кросплатформенный веб-интерфейс для формирования параметров генерации движения и модуль взаимодействия аппаратного пульта с данной подсистемой.

Данная работа состоит из 63 страниц, включает 32 рисунка и 2 таблицы.

## ABSTRACT

Creating hardware software for controlling an insectomorphic robot is proposed. A subsystem for exchanging data with the FRUND program, a cross-platform web interface for formation of motion generation parameters, and a module for interacting a hardware remote control joystick with this subsystem have been implemented.

This paper consists of 63 pages and includes 32 figures and 2 tables.

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ . . . . .	7
1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ СИСТЕМЫ УПРАВЛЕНИЯ ШАГАЮЩИМИ РОБОТАМИ . . . . .	9
1.1 Шагающие роботы . . . . .	9
1.2 Инсектоморфные роботы . . . . .	12
1.3 Методы управления роботами. . . . .	13
1.3.1 Датчики на шагающих роботах . . . . .	13
1.3.2 Типы походки и баланс . . . . .	14
1.4 Методы управления инсектоморфными роботами . . . . .	17
1.4.1 Координация ног . . . . .	17
1.4.2 Механизм ног . . . . .	18
1.4.3 Кинематическая модель шестиного шагающего робота . . . . .	21
1.4.4 Выводы по главе. . . . .	23
2 ПРОЕКТИРОВАНИЕ СИСТЕМЫ УПРАВЛЕНИЯ РОБОТОМ- ГЕКСАПОДОМ. . . . .	24
2.1 Описание системы управления . . . . .	24
2.2 Общая структура системы управления . . . . .	26
2.3 Проектирование системы управления роботом-гексаподом . . . . .	27
2.3.1 Спецификация команд управления . . . . .	27
2.3.2 Постановка и анализ требований . . . . .	29
2.3.3 Проектирование архитектуры системы управления. . . . .	30
2.4 Обзор решения-аналога . . . . .	33
2.4.1 Архитектура и сценарий использования . . . . .	34
2.4.2 Анализ и сравнение с предложенным решением . . . . .	36
2.5 Выводы по главе . . . . .	36
3 РЕАЛИЗАЦИЯ СИСТЕМЫ УПРАВЛЕНИЯ РОБОТОМ ГЕКСАПОДОМ. . . . .	40
3.1 Sketch . . . . .	40
3.2 Front-end и VueJS . . . . .	41
3.3 Browser-Sync. . . . .	43

3.4 ROS . . . . .	44
3.5 Rosbridge . . . . .	50
3.6 GAZEBO . . . . .	51
3.7 Выводы по главе . . . . .	54
4 ТЕСТИРОВАНИЕ СИСТЕМЫ УПРАВЛЕНИЯ РОБОТОМ-ГЕКСАПОДОМ. . . . .	55
4.1 Выводы по главе . . . . .	59
ЗАКЛЮЧЕНИЕ . . . . .	60
СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ . . . . .	61

## ВВЕДЕНИЕ

В настоящее время активно ведутся исследования и разработки в области создания и использования инсектоморфных роботов, а также реализация систем управления и ориентирования роботов данного типа в пространстве. Данные тенденции обусловлены тем, что этот вид роботов является более проходимым, по сравнению с другими, которые, как правило, оборудуются колесными и гусеничными движителями.

Как можно заметить, рельеф земли, в основном, представляет собой неровные поверхности, соответственно можно выделить следующие преимущества инсектоморфных роботов:

- возможность самостоятельно выбирать точку касания с поверхностью;
- адаптивное движение по отношению к поверхности.

Стоит отметить, что не каждого шагающего робота можно считать "настоящим" роботом, так как в наше время создано множество различных моделей, которые шагают сугубо номинально, используя лишь циклические походки и циклическую генерацию движения, однако этого недостаточно. Для настоящей ходьбы робот должен быть оснащен различными инерционными датчиками, иметь возможность измерения неровности поверхности по которой производится движение, расстояние до нее, а также уметь подстраивать свою походку под эти неровности, обеспечивая возможность регулируемого взаимодействия с опорной поверхностью.

Такое стало возможным только в последнее время с развитием вычислительной техники. Этому способствовало появление компьютеров, обладающих достаточными вычислительными мощностями и ресурсами и при всем при том в сравнительно компактном корпусе, что позволяет устанавливать их непосредственно на робота и в режиме реального времени обрабатывать показания, которые принимаются с датчиков и за достаточно маленький период времени изменять положение приводов, основываясь на этих показаниях.

Поэтому актуальной является работа по разработке системы для ре-

шения задачи удаленного управления инсектоморфным роботом в режиме реального времени с использованием генерации движений за помощью математического пакета ФРУНД.

Целью данной выпускной бакалаврской работы является разработка ПО для обеспечения возможности передачи параметров, необходимых для генерации движения непосредственно во ФРУНД, а также интегрирование разработанного ПО, как часть системы управления инсектоморфным роботом.

Для достижения данной цели необходимо решить следующие задачи:

- 1) анализ предметной области существующих способов управления роботами такого типа;
- 2) постановка задачи проектирования подсистемы формирования параметров генерации движения;
- 3) реализация подсистемы формирования параметров управления и её интеграция в существующей системе управления роботом;
- 4) проведение экспериментов и оценка результатов работы.

В первой главе настоящей работы приведен обзор различных видов роботов и методов управления ими.

Во второй главе рассматривается проектирование системы управления роботом-гексаподом, сравнение с аналогом и постановка требований к реализации.

В третьей главе рассматривается этап реализации системы управлением роботом гексаподом и необходимые компоненты данной системы.

В четвертой главе проводятся тесты системы управления на соответствие поставленным требованиям.

В разделе "Выводы" подводятся итоги создания подсистемы дистанционного управления инсектоморфным роботом и рассматриваются результаты экспериментов.

# 1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ СИСТЕМЫ УПРАВЛЕНИЯ ШАГАЮЩИМИ РОБОТАМИ

В этой главе приводится обзор существующих на данный момент видов роботов, которые имеют непосредственное отношение к данной работе. Рассматриваются как шагающие роботы в целом, так и инсектоморфные, являющихся подвидом шагающих. В конце главы также рассматриваются методы управления такими роботами.

## 1.1 Шагающие роботы

Шагающие роботы являются важной альтернативой роботам, использующим колесные и гусеничные движители, поскольку большая часть земной площади редко представляет из себя даже грунтовую дорогу. Таким образом, если ездящие роботы более специализированы и лучше приспособлены для плоских поверхностей - они могут двигаться быстрее и лучше ориентироваться в пространстве, используя различные системы навигации, то шагающие роботы могут использоваться в более разнообразных условиях. Такие роботы следуя природе, имеют возможность перемещаться по пересеченной местности или даже подниматься по лестнице или преодолевать препятствия в стандартной бытовой ситуации, на что не способно большинство ездящих роботов. Также ноги шагающего робота наносят значительно меньший ущерб местности, чем колесные или гусеничные машины. Кроме того, положение тела робота может быть изменено, при удержании ног на земле, таким образом, добавляется еще одна степень свободы для выполнения различных задач. Пример перемещения робота по пересеченной поверхности приведен на рисунке 1 на котором отражен принцип работы модуля motion planner, отвечающего за вычисление последовательности действий, необходимых для перемещения ног от исходной точки к конечной, с учетом рельефа поверхности.

Роботы с шестью или более ногами обладают преимуществом стабильности. В типичной манере ходьбы шестиногого робота три ноги всегда находятся

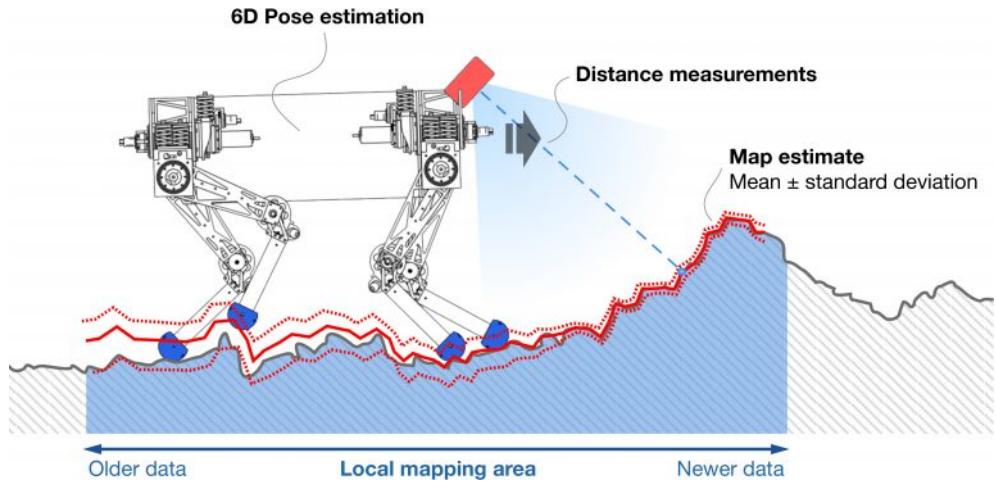


Рисунок 1 – Работа motion planner на пересеченной местности

дятся на земле, а три ноги движутся. Это дает статический баланс при ходьбе, при условии, что центр масс робота находится внутри треугольника, образованного тремя ногами на земле. Четвероногих роботов значительно сложнее сбалансируировать, но они все же довольно просты по сравнению с динамикой двуногих роботов. Двуногих роботов сложнее всего сбалансируовать: во время ходьбы только одна нога на земле и одна нога в воздухе. Статический баланс для двуногих роботов может быть достигнут, если ноги робота относительно велики и области контакта с землей обеих ног перекрываются. Тем не менее, это не относится к подобным человеку роботам-androидам, которые требуют динамического баланса для ходьбы.

В целом, когда дело доходит до моделей с двумя ногами, мы получаем роботов, которые напоминают то, о чем думает большинство людей, когда слышат термин «робот». Это двуногие роботы, которые часто называют «человекоподобными роботами» или «роботами-андроидами» из-за их сходства с людьми. Один из антропоморфных роботов, над которым сейчас ведутся работы по обучению различным автономным задачам, располагается в Волгоградском государственном техническом университете на факультете электроники и вычислительной техники. Это модель AR-600E (рис. 2) из серии AR60x, разрабатываемая компанией НПО "Андроидная техника" в г.Магнитогорск.

Стоит отметить, что именно структура ног отделяет локомоцию тела от движения ног при преодолении препятствий. В результате тело может поддерживать равновесие, что приводит к хорошей адаптации большинства распространенных ландшафтов. Поэтому в последнее время именно шагающие

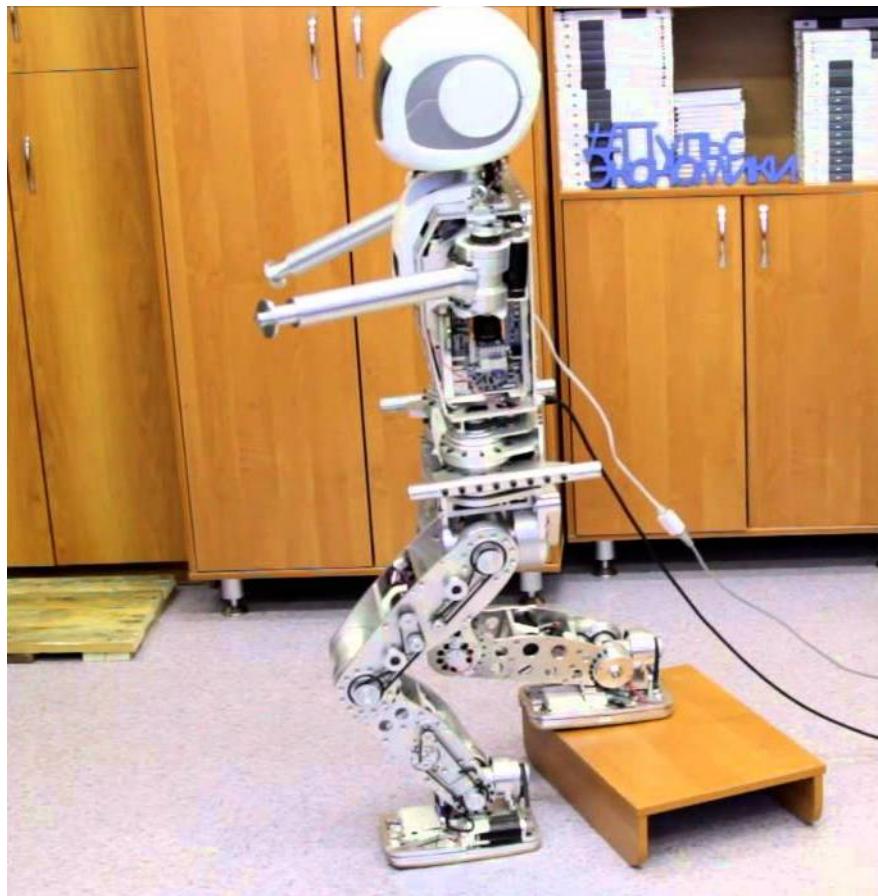


Рисунок 2 – Антропоморфный робот AR-600E

роботы являются наиболее приоритетными в исследованиях, проводимыми в области роботов. Именно ноги являются ключевыми частями конструкции шагающего робота.

Применение шагающих роботов представляет собой довольно обширную область. Это могут быть как гражданские цели, такие как доставка грузов, так и военные и медицинские. Достигнув оптимального подхода с помощью роботизированных механизмов, шагающие роботы смогут выполнить опасные задачи в армии, помочь некоторым людям, использующим инвалидные коляски, приобретать «ходячие» способности. В плане медицины можно получить довольно значительные улучшения, возможно, с протезами, которые работают намного лучше, чем существующие технологии, или даже создание экзоскелетов, которые могли бы позволить человеку с ограниченными двигательными способностями эффективно ходить. В условиях особой опасности, шагающие роботы могут пригодиться военным, полицейским и спасателям. К примеру, можно проводить операции по обнаружению мин и обезвреживанию в суровых условиях с множеством препятствий, где обычные транспортные средства не могут справиться с задачей. Уникальной характе-

ристикой пешеходных транспортных средств является способность двигаться с дискретным размещением ног, что позволяет роботу избегать наступления на мины или другие деликатные объекты.

## 1.2 Инсектоморфные роботы

Инсектоморфными называется такой вид шагающих роботов, у которых по аналогии с насекомыми (англ. insect - насекомое) располагается три пары конечностей. Фактически, элементарное движение шестиногого шагающего робота может быть достигнуто простым переключением поддержки робота между набором ножек, которые образуют треугольник. Кроме того, для обеспечения статического хождения координация шести ножек может быть осуществлена путем наложения подходящего запаса устойчивости между проекцией грунта центра тяжести робота и многоугольником между опорными лапами.

Другой подход к дизайну шестиногих шагающих роботов можно получить, обратившись к биологическим системам и, таким образом, разработать биологически вдохновленный дизайн робота. Фактически, согласно "техническому замыслу", биологическое вдохновение может быть лишь тривиальным наблюдением того, что некоторые насекомые используют шесть ног, которые полезны для получения устойчивой опоры во время ходьбы, в то время как "биологический замысел" означает подражание движению конкретного вида насекомых в каждой детали. В общем, насекомые ходят на нескольких скоростях с различными походками, обладающими свойством статической устойчивости, но одной из ключевых характеристик управления передвижением является распределение.

Таким образом, в отличие от простого управления переключением "технического дизайна", управление распределенной походкой следует рассматривать в соответствии с "биологическим дизайном" шестиногого шагающего робота, который пытается эмулировать передвижение конкретного насекомого. Другими словами, вместо централизованной системы управления движением робота можно считать, что различные локальные контроллеры ног обеспечивают управление распределенной походкой.

В ВолгГТУ на кафедре электронных вычислительных машин и систем

уже был разработан и собран опытный образец инсектоморфного робота гексапода (рис. 3), реализация управления для которого рассматривается далее в данной работе. Также ведутся работы в области создания более продвинутой модели гексапода, и их оснащение техническим зрением и манипулятором.

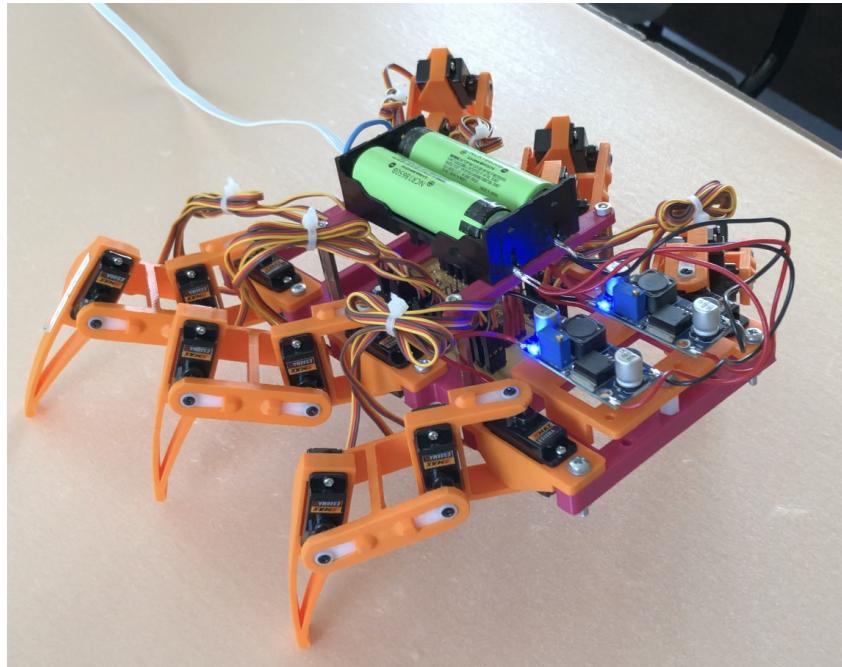


Рисунок 3 – Опытная модель гексапода

### 1.3 Методы управления роботами

Решение задачи управления роботами является одной из основополагающих. В общем случае, колесные и гусеничные роботы являются довольно простыми при ручном управлении. Для этого обычно применяются различные пульты дистанционного управления или специальное ПО для компьютера. Простота заключается в том, что как правило, для управления достаточно лишь функции движения прямо, в обратном направлении, а также функций поворота. Все это происходит либо на плоскости, либо в условиях легкого бездорожья. Однако, когда дело касается шагающих роботов все усложняется необходимостью предусмотреть системы обратной связи и алгоритмы для удержания равновесия.

#### 1.3.1 Датчики на шагающих роботах

Датчики обратной связи являются основой динамического баланса и ходьбы в целом. Вот почему уделяется особое внимание выбору подходящих

датчиков и их использованию для управления роботом. С другой стороны, существует целый ряд роботов, которые не используют какие-либо датчики на всех и полагаются исключительно на стабильную механическую конструкцию с большой площадью опоры для балансировки и ходьбы.

Рассмотрим датчики, которые применяются в роботах:

- Инфракрасные датчики приближения на ногах дают обратную связь о том, соприкасается ли они с землей;
- Датчики ускорения в двух осях для измерения динамических сил на роботе, чтобы сбалансируировать его;
- Пьезо гироскопы используются в качестве альтернативы датчикам ускорения, которые подвержены высокочастотному сервошуму. Поскольку гироскопы только возвращают изменение ускорения, их значения должны быть интегрированы для сохранения общей ориентации;
- Инклинометры используются для поддержки гироскопов. Хотя инклинометры нельзя использовать по отдельности из-за их временной задержки, их можно использовать для устранения смещения датчиков и ошибок интеграции гироскопов;
- Инфракрасные PSD размещенные в направлениях вперед, влево и вправо, робот может распознавать окружающие препятствия.
- Цифровая камера может использоваться двумя способами: либо для поддержки балансировки, либо для обнаружения объектов, пешеходных дорожек и т.д.

### 1.3.2 Типы походки и баланс

Есть два типа походки, используя:

- Статический баланс – центр масс робота всегда находится в пределах области поддержки его ноги на земле или объединенной области поддержки его двух ног, если обе ноги находятся на земле.
- Динамический баланс - центр тяжести робота может находиться за пределами зоны поддержки его ног во время фазы его походки.

Также стоит выделить следующие параметры походки:

- 1) Длина шага;
- 2) Высота шага;

- 3) Скорость ходьбы;
- 4) Угол наклона туловища;
- 5) Максимальный угол наклона.

Затем параметры походки обновляются в режиме реального времени в зависимости от датчиков робота. Показания датчика тока сравниваются с требуемыми показаниями датчика в каждый момент времени походки. Различия между текущим и желаемым показаниями датчика приведут к немедленной адаптации параметров к модели походки.

Модели походки, основанные на статическом равновесии, не очень эффективны. Они требуют больших площадей ног и возможны только относительно медленные походки, чтобы поддерживать динамические силы на низком уровне. Механизмы ходьбы с динамическим балансом, с другой стороны, позволяют создавать роботов с более мелкими ступнями, даже ступнями, которые имеют только одну точку контакта и могут использоваться для гораздо более быстрой ходьбы или даже для бега.

Как было определено ранее, динамическое равновесие означает, что, по крайней мере, на некоторых этапах походки робота его центр масс не поддерживается его ногой. Игнорирование любых динамических сил и моментов означает, что робот упадет, если в реальном времени не будет предпринято противодействие. Существует ряд различных подходов к динамической ходьбе, описанных ниже:

- 1) Нулевой момент это один из стандартных методов динамического баланса и опубликован в ряде статей. Реализация этого метода требует знания всех динамических сил на теле робота, а также всех моментов между ногой и лодыжкой робота. Эти данные можно определить с помощью акселерометров или гироскопов на теле робота, а также датчиков давления в ногах робота или датчиков крутящего момента на лодыжках робота. При всех известных силах контакта и всех динамических силах робота можно рассчитать «точку нулевого момента», которая является динамическим эквивалентом статического центра масс. Если она находится в зоне поддержки ноги робота (или обеих ног) на земле, то робот находится в динамическом равновесии. В противном случае необходимо предпринять корректирующие действия, изменив положение тела робота, чтобы избежать его падения;

- 2) Перевернутый маятник - шагающего робота можно смоделировать как перевернутый маятник. Динамического равновесия можно достичь, постоянно отслеживая ускорение робота и адаптируя соответствующие движения ног;
- 3) Нейронные сети могут быть использованы для достижения динамического баланса. Конечно, этот подход все еще требует обратной связи с датчиками, как и в других подходах;
- 4) Генетические алгоритмы. Популяция виртуальных роботов создается с изначально случайными настройками управления. Роботы с лучшими характеристиками воспроизводятся с использованием генетических алгоритмов для следующего поколения.

Такой подход на практике требует наличия системы симуляции механики для оценки производительности каждого отдельного робота, и даже в этом случае для достижения хороших характеристик ходьбы требуется несколько дней использования процессора. Основной проблемой здесь является возможность передачи результатов моделирования обратно физическому роботу;

- 5) PID регулятор используется для управления наклоном робота, аналогично случаю статического баланса. Однако нет необходимости заставлять робота стоять прямо. Вместо этого на этапе обучения записывается желаемый передний и боковой наклон тела робота на всех этапах его походки. Позже, при контролировании походки, необходимо добиться этого смещения переднего и бокового наклона с помощью PID регулятора. Для достижения такого наклона можно устанавливать следующие параметры:

- Длина шага;
- Высота шага;
- Скорость ходьбы;
- Наклон торса вперед;
- Максимальный боковой сдвиг.

- 6) Искусственный горизонт. В этом подходе используются не кинетические датчики других подходов, а монокулярная камера в градациях серого. В простой версии черная линия на белом фоне помещается в поле зрения робота. Затем мы можем измерить ориентацию робота путем

изменения положения и ориентации линии на изображении. Например, линия будет двигаться к вершине, если робот падает вперед, она будет наклонена под углом, если робот наклоняется влево, и т.д. С более мощным контроллером для обработки изображений, тот же принцип может быть применен даже без необходимости искусственного горизонта. Пока на заднем плане достаточно текстуры, для определения движения робота можно использовать общий оптический поток.

## 1.4 Методы управления инсектоморфными роботами

Как уже писалось ранее, шестиногий робот в его правильном варианте исполнения должен быть биологически вдохновленным. Эмуляция локомоций палочниковых насекомых должна выполняться посредством прямой ходьбы на разных скоростях и ходьбы по кривым или в разных направлениях. Поэтому стоит рассмотреть некоторые технические аспекты ходьбы для построения математической модели.

### 1.4.1 Координация ног

Основываясь на рисунке 4, система отсчета  $G'(x'_G \ y'_G \ z'_G)$  имеющая начало в  $G'$ , совпадающая с проекцией на землю центра масс  $G$  тела шестиножки и шести систем отсчета  $O_{Si}(x_{Si} \ y_{Si} \ z_{Si})$  для  $i = 1, \dots, 6$ , была выбрана для анализа и оптимизации движения каждого конца ноги с целью обеспечения подходящей статической устойчивости при ходьбе.

Таким образом, вкратце, движение каждого конца ноги можно выразить как функцию параметров  ${}^{Si}p_{ix}$  и  $s_i$ , где  ${}^{Si}p_{ix}$  дает положение конца ноги в  $O_{Si}(x_{Si} \ y_{Si} \ z_{Si})$  вдоль оси  $x$  для фазы опоры, а  $s_i \in \{0; 1\}$  указывает состояние каждого конца ноги, т.е.  $s_i = 0$  используется для фазы переноса, а  $s_i = 1$  для фазы опоры, которые выполняются в диапазоне  $[PEP_i, AEP_i]$ , где  $PEP_i$  – это заднее крайнее положение, а  $AEP_i$  – переднее крайнее положение каждой конечности. В частности,  $L$  – это номинальное расстояние между  $PEP_0$  и  $AEP_0$ . Траектория каждого конца ноги во время фазы переноса определяется с учетом времени начала и окончания фазы опоры.

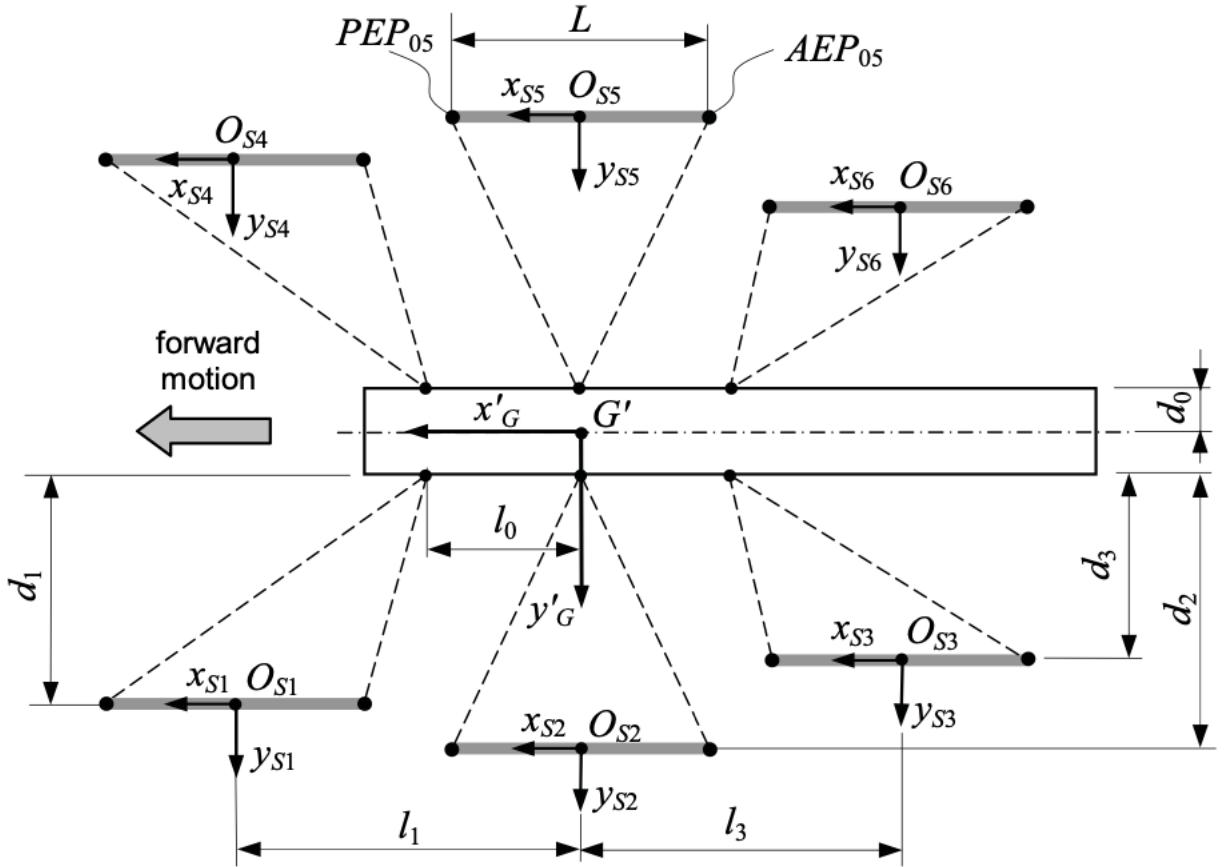


Рисунок 4 – Общая схема гексапода

#### 1.4.2 Механизм ног

Кинематическая цепочка с тремя оборотами для каждой ноги используется в целях имитирования структуры ноги палочника, а именно соха, femur и tibia как показано на рисунке 5. Непосредственный кинематический анализ каждого механизма ноги формируется между подвижной системой координат  $O_{Ti}(x_{Ti} \ y_{Ti} \ z_{Ti})$  тела tibia и системой координат  $O_{0i}(x_{0i} \ y_{0i} \ z_{0i})$ , которая рассматривается как неподвижная рама, прежде чем она будет соединена с корпусом робота, чтобы сформулировать общую модель гексапода, показанного на рисунке 6.

В частности общая матрица преобразования  $\mathbf{M}_{Ti}^{0i}$  между подвижной системой координат  $O_{Ti}(x_{Ti} \ y_{Ti} \ z_{Ti})$  и неподвижной системой координат определяется как:

$$\mathbf{M}_{Ti}^{0i}(\vartheta_{1i}, \vartheta_{2i}, \vartheta_{3i}) = \begin{bmatrix} r_{11} & r_{12} & r_{13} & {}^{0i}p_{ix} \\ r_{21} & r_{22} & r_{23} & {}^{0i}p_{iy} \\ r_{31} & r_{32} & r_{33} & {}^{0i}p_{iz} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

Эта матрица получается как произведение между четырьмя матрицами преобразования, которые связывают подвижную систему tibia с тремя типичными опорными звеньями на поворотных суставах (joints) механизма ноги.

Таким образом, каждая запись  $r_{jk}$  в  $\mathbf{M}_{Ti}^{0i}$  для  $j, k = 1, 2, 3$  и декартовой компоненты положения вектора  $\mathbf{p}_i$  в системе координат  $O_{0i}(x_{0i} \ y_{0i} \ z_{0i})$  задается выражением:

$$\begin{aligned}
r_{11} &= c\alpha_0 s\vartheta_{1i}; \quad r_{21} = -c\vartheta_{1i}; \quad r_{31} = -s\vartheta_{1i} s\alpha_0 \\
r_{12} &= s\vartheta_{3i} (s\alpha_0 s\vartheta_{2i} - c\alpha_0 c\vartheta_{1i} c\vartheta_{2i}) - c\vartheta_{3i} (c\alpha_0 c\vartheta_{1i} s\vartheta_{2i} + s\alpha_0 c\vartheta_{2i}) \\
r_{22} &= -s\vartheta_{1i} c\vartheta_{2i} s\vartheta_{3i} - s\vartheta_{1i} s\vartheta_{2i} c\vartheta_{3i} \\
r_{32} &= s\vartheta_{3i} (c\alpha_0 s\vartheta_{2i} + s\alpha_0 c\vartheta_{1i} c\vartheta_{2i}) + c\vartheta_{3i} (s\alpha_0 c\vartheta_{1i} s\vartheta_{2i} + c\alpha_0 c\vartheta_{2i}) \\
r_{13} &= c\vartheta_{3i} (c\alpha_0 c\vartheta_{1i} c\vartheta_{2i} - s\alpha_0 s\vartheta_{2i}) - s\vartheta_{3i} (c\alpha_0 c\vartheta_{1i} s\vartheta_{2i} + s\alpha_0 c\vartheta_{2i}) \\
r_{23} &= s\vartheta_{1i} c\vartheta_{2i} c\vartheta_{3i} - s\vartheta_{1i} s\vartheta_{2i} s\vartheta_{3i} \\
r_{33} &= -c\vartheta_{3i} (s\alpha_0 c\vartheta_{1i} c\vartheta_{2i} + c\alpha_0 s\vartheta_{2i}) + s\vartheta_{3i} (s\alpha_0 c\vartheta_{1i} s\vartheta_{2i} - c\alpha_0 c\vartheta_{2i}) \\
{}^0 p_{ix}^i &= [c\vartheta_{3i}^2 (c\alpha_0 c\vartheta_{1i}^2 c\vartheta_{2i} - s\alpha_0 s\vartheta_{2i}) - s\vartheta_{3i} (c\alpha_0 c\vartheta_{1i} s\vartheta_{2i} + s\alpha_0 c\vartheta_{2i})] a_3 + \\
&\quad + (c\alpha_0 c\vartheta_{1i} c\vartheta_{2i} - s\alpha_0 s\vartheta_{2i}) a_2 + c\alpha_0 c\vartheta_{1i} a_1 \\
{}^0 p_{iy}^i &= a_3 (s\vartheta_{1i} c\vartheta_{2i} c\vartheta_{3i} - s\vartheta_{1i} s\vartheta_{2i} s\vartheta_{3i}) + (s\vartheta_{1i} c\vartheta_{2i}) a_2 + s\vartheta_{1i} a_1 \\
{}^0 p_{iz}^i &= [-c\vartheta_{3i} (s\alpha_0 c\vartheta_{1i} c\vartheta_{2i} + c\alpha_0 s\vartheta_{2i}) + s\vartheta_{3i} (s\alpha_0 c\vartheta_{1i} s\vartheta_{2i} - c\alpha_0 c\vartheta_{2i})] a_3 + \\
&\quad - (s\alpha_0 c\vartheta_{1i} c\vartheta_{2i} - c\alpha_0 s\vartheta_{2i}) a_2 - s\alpha_0 c\vartheta_{1i} a_1,
\end{aligned} \tag{2}$$

где  $\vartheta_{1i}$ ,  $\vartheta_{2i}$  и  $\vartheta_{3i}$  переменные углов суставов каждой из ножек ( $i = 1, \dots, 6$ ),  $\alpha_0$  – угол первой оси соединения с осью  $z_{0i}$ , а  $a_1, a_2, a_3$  – длины соха, femur и tibia соответственно.

Обратный кинематический анализ механизма ног формируется при помощи алгебраического похода. Таким образом, когда декартовы компоненты вектора положения  $\mathbf{p}_i$  заданы в системе координат  $O_{Fi}(x_{Fi} \ y_{Fi} \ z_{Fi})$  переменные углы соединения  $\vartheta_{1i}, \vartheta_{2i}, \vartheta_{3i} (i = 1, \dots, 6)$  можно выразить как:

$$\vartheta_{1i} = atan2({}^F p_{iy}^i, {}^F p_{ix}^i) \tag{3}$$

и

$$\vartheta_{3i} = atan2(s\vartheta_{3i}, c\vartheta_{3i}) \tag{4}$$

где

$$\begin{aligned} c\vartheta_{3i} &= \frac{\left({}^F_i p_{ix}\right)^2 + \left({}^F_i p_{iy}\right)^2 + \left({}^F_i p_{iz}\right)^2 + a_1^2 - 2a_1\sqrt{\left({}^F_i p_{ix}\right)^2 + \left({}^F_i p_{iy}\right)^2} - a_2^2 - a_3^2}{2a_2a_3}, \\ s\vartheta_{3i} &= \pm\sqrt{1 - c^2\vartheta_{3i}} \end{aligned} \quad (5)$$

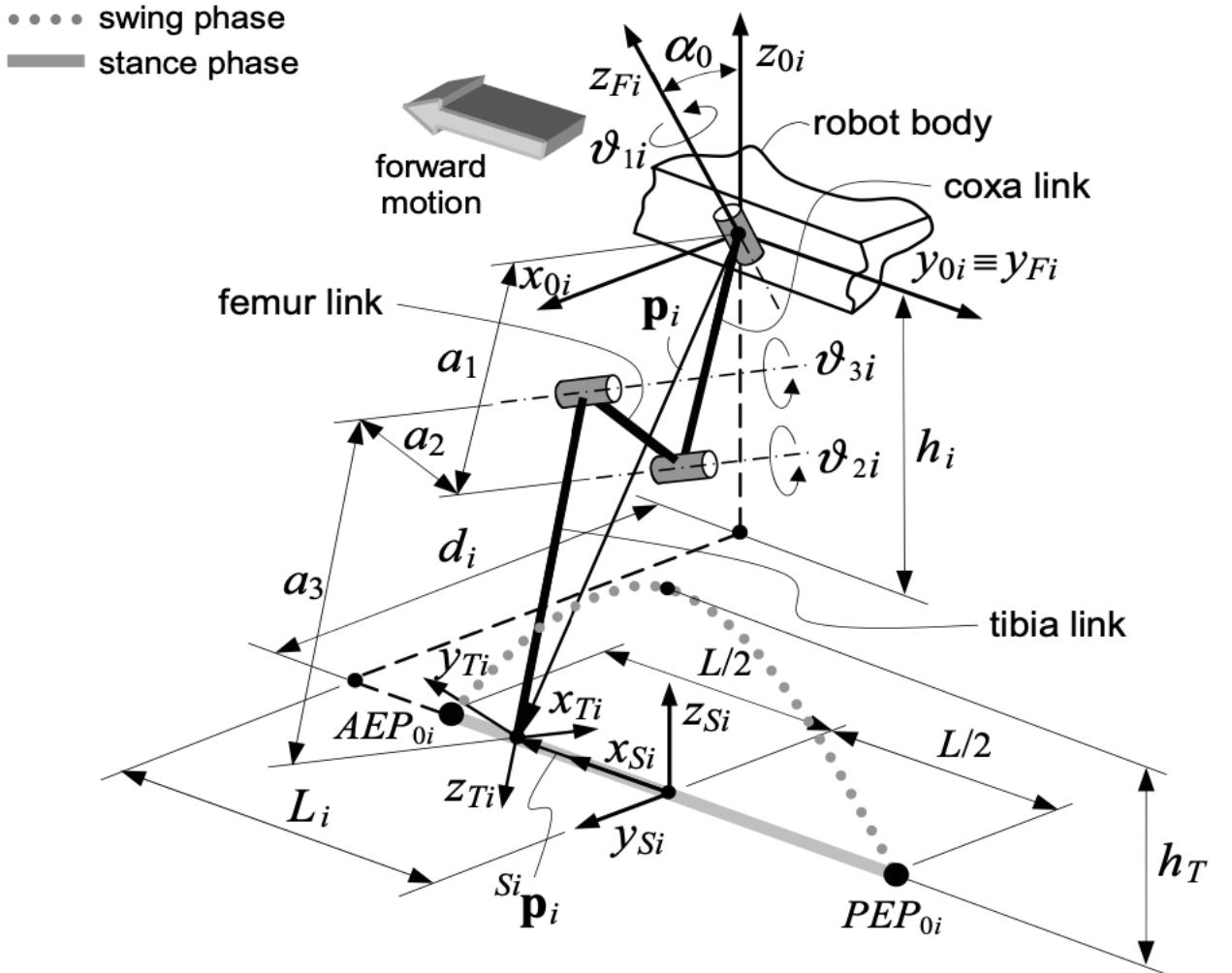


Рисунок 5 – Механизм 3R ноги шестиногого шагающего робота  
И в свою очередь:

$$v_{2i} = \text{atan2}(s\vartheta_{2i}, c\vartheta_{2i}), \quad (6)$$

где

$$\begin{aligned} s\vartheta_{2i} &= -\frac{a_3s\vartheta_{3i}(\sqrt{\left({}^F_i p_{ix}\right)^2 + \left({}^F_i p_{iy}\right)^2}) + {}^F_i p_{iz}(a_2 + a_3c\vartheta_{3i}^2)}{a_2^2 + a_3^2 + 2a_2a_3c\vartheta_{3i}} \\ c\vartheta_{2i} &= -\frac{{}^F_i p_{iz} + s\vartheta_{2i}(a_2 + a_3c\vartheta_{3i}^2)}{a_3s\vartheta_{3i}} \end{aligned} \quad (7)$$

Таким образом, уравнения (1-7) позволяют сформулировать общую кинематическую модель шестиногого робота, что описывается ниже.

### 1.4.3 Кинематическая модель шестиногого шагающего робота

Полагаясь на рисунок 5 и 6, кинематическая модель шестиногого шагающего робота сформулирована посредством прямого кинематического анализа между движемой системой координат  $O_{Ti}(x_{Ti} \ y_{Ti} \ z_{Ti})$  звена tibia и инерционной системой координат  $O(X \ Y \ Z)$ .

В общем, шестиногий шагающий робот имеет 24 степени свободы, где 18 степеней свободы задаются значениями  $\vartheta_{1i}, \vartheta_{2i}, \vartheta_{3i}$  ( $i = 1, \dots, 6$ ) для шести механизмов ног 3R, а 6 степеней свободы определяются как тело робота, которые уменьшаются в этом случае только на 1 градус – это дано  $X_G$ , чтобы рассмотреть чистый переход тела робота вдоль оси  $X$ .

Таким образом, уравнение движения  $X_G(t)$  тела робота назначается в качестве входных данных для предложенного алгоритма, в то время как  $\vartheta_{1i}(t), \vartheta_{2i}(t), \vartheta_{3i}(t)$  для  $i = 1, \dots, 6$  выражены посредством обратного кинематического анализа шести механизмов ног 3R, когда дано уравнение движения каждого конца ноги и задана их форма траектории во время фазы переноса.

В частности, матрица преобразования  $\mathbf{M}_G$  системы координат  $G(x_G \ y_G \ z_G)$  на корпусе робота относительно инерционной системы координат  $O(X \ Y \ Z)$  выражается как:

$$\mathbf{M}_G(X_G) = \begin{bmatrix} 1 & 0 & 0 & p_{Gx} \\ 0 & 1 & 0 & p_{Gy} \\ 0 & 0 & 1 & p_{Gz} \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (8)$$

где  $p_{GX} = X_G$ ,  $p_{GY} = 0$  и  $p_{GZ} = h_G$ .

Матрица преобразования  $\mathbf{M}_{Bi}^G$  системы координат  $O_{Bi}(x_{Bi} \ y_{Bi} \ z_{Bi})$  на корпусе робота относительно системы координат  $G(x_G \ y_G \ z_G)$  выражается

как:

$$\mathbf{M}_{Bi}^G = \begin{cases} \begin{bmatrix} 0 & 1 & 0 & d_0 \\ -1 & 0 & 0 & l_i \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, & i = 1, 2, 3 \\ \begin{bmatrix} 0 & -1 & 0 & d_0 \\ 1 & 0 & 0 & -l_i \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, & i = 4, 5, 6 \end{cases}, \quad (9)$$

где  $l_1 = l_4 = -l_0, l_2 = l_5 = 0, l_3 = l_6 = l_0$ .

Поэтому прямая кинематическая функция шагающего робота определяется как:

$$\mathbf{M}_{Ti}(X_G, \vartheta_{1i}, \vartheta_{2i}, \vartheta_{3i}) = \mathbf{M}_G(X_G) \mathbf{M}_{Bi}^G \mathbf{M}_{0i}^{Bi} \mathbf{M}_{Ti}^{0i}(\vartheta_{1i}, \vartheta_{2i}, \vartheta_{3i}), \quad (10)$$

где  $\mathbf{M}_{0i}^{Bi} = \mathbf{I}$ , является **I** единичной матрицей. Углы суставов (joints) механизмов ног получены за помощью обратного кинематического анализа. Кроме того матрица преобразования  $\mathbf{M}_{Si}^{Bi}$  имеет вид:

$$\mathbf{M}_{Bi}^{Bi} = \begin{cases} \begin{bmatrix} 0 & -1 & 0 & L_i \\ 1 & 0 & 0 & d_i \\ 0 & 0 & 1 & -h_i \\ 0 & 0 & 0 & 1 \end{bmatrix}, & i = 1, 2, 3 \\ \begin{bmatrix} 0 & 1 & 0 & L_{i-3} \\ -1 & 0 & 0 & -d_{i-3} \\ 0 & 0 & 1 & -h_i \\ 0 & 0 & 0 & 1 \end{bmatrix}, & i = 4, 5, 6 \end{cases}, \quad (11)$$

где  $L_1 = l_1 - l_0, L_2 = 0$  и  $L_3 = l_3 - l_0$  с  $L_i$ , показанной на рисунке 5. Наконец,

положение каждого конца ноги в системе координат  $O_{Fi}(x_{Fi} \ y_{Fi} \ z_{Fi})$

$${}^{F_i} \mathbf{p}_i(t) = \mathbf{M}_{0i}^{F_i} \mathbf{M}_{Bi}^{0i} \mathbf{M}_{Si}^{Bi} {}^{S_i} \mathbf{p}_i(t), \quad (12)$$

где зная матрицу  $\mathbf{M}_{Si}^{Bi}$  легко можно получить, зная угол  $\alpha_0$ .

Следовательно, подставляя декартовы компоненты  ${}^{F_i} \mathbf{p}_i(t)$  в уравнения (3), (5) и (7) могут быть получены углы суставов  $\vartheta_{1i}, \vartheta_{2i}, \vartheta_{3i}$  ( $i = 1, \dots, 6$ ).

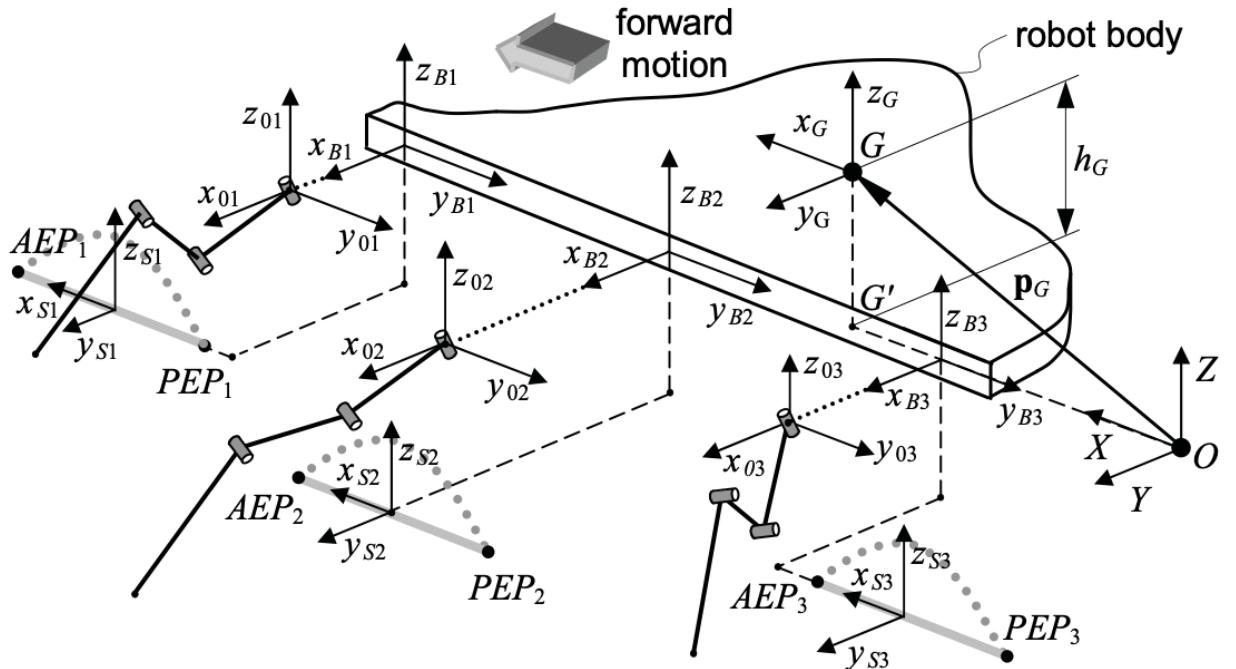


Рисунок 6 – Кинематическая схема шестиногого шагающего робота

#### 1.4.4 Выводы по главе

Был проведен обзор шагающих роботов, а также инсектоморфных, как подвида. Рассмотрены главные достоинства шагающих роботов перед колесными и гусеничными роботами. Приведен подробный анализ существующих методов управления шагающими роботами, а также разбор кинематической модели инсектоморфного робота, построенного по аналогии с насекомым палочником.

## 2 ПРОЕКТИРОВАНИЕ СИСТЕМЫ УПРАВЛЕНИЯ РОБОТОМ-ГЕКСАПОДОМ

В данной главе рассматривается этап проектирования управления гексаподом, структура системы управления, подбор технических средств для его реализации, а так же постановка требований, которым должно удовлетворять конечное решение.

### 2.1 Описание системы управления

На борту гексапода установлен 32 битный ARM-контроллер STM32F100RET6B, который непосредственно, путем подачи необходимого ШИМ – сигнала управляет сервоприводами, установленными в ногах. Принципиальная схема устройства изображена на рисунке 7

ШИМ – это одно из основных средств, с помощью которых микроконтроллеры управляют аналоговыми устройствами, такими как двигатели с регулируемой скоростью, регулируемые источники света, исполнительные механизмы и динамики. Однако ШИМ не является истинным аналоговым выходом. ШИМ "подделывает" аналогичный результат, применяя мощность в импульсах или коротких импульсах регулируемого напряжения.

Таким образом, подавая ШИМ сигнал различной скважности с контроллера на сервоприводы можно устанавливать определенный угол и тем самым управлять конечностями робота.

Для расчетов и генерации походки используется математический пакет ФРУНД – программная система формирования решений уравнений нелинейной динамики. Основной задачей данного пакета является моделирование динамических процессов в машинах и конструкциях. По созданному пользователем описанию расчетной схемы есть возможность создания уравнений математической модели динамики движения исследуемой конструкции, а также генерации программы интегрирования данных уравнений. Решение, обработка и вывод результатов предоставляется в удобной для пользователя форме.

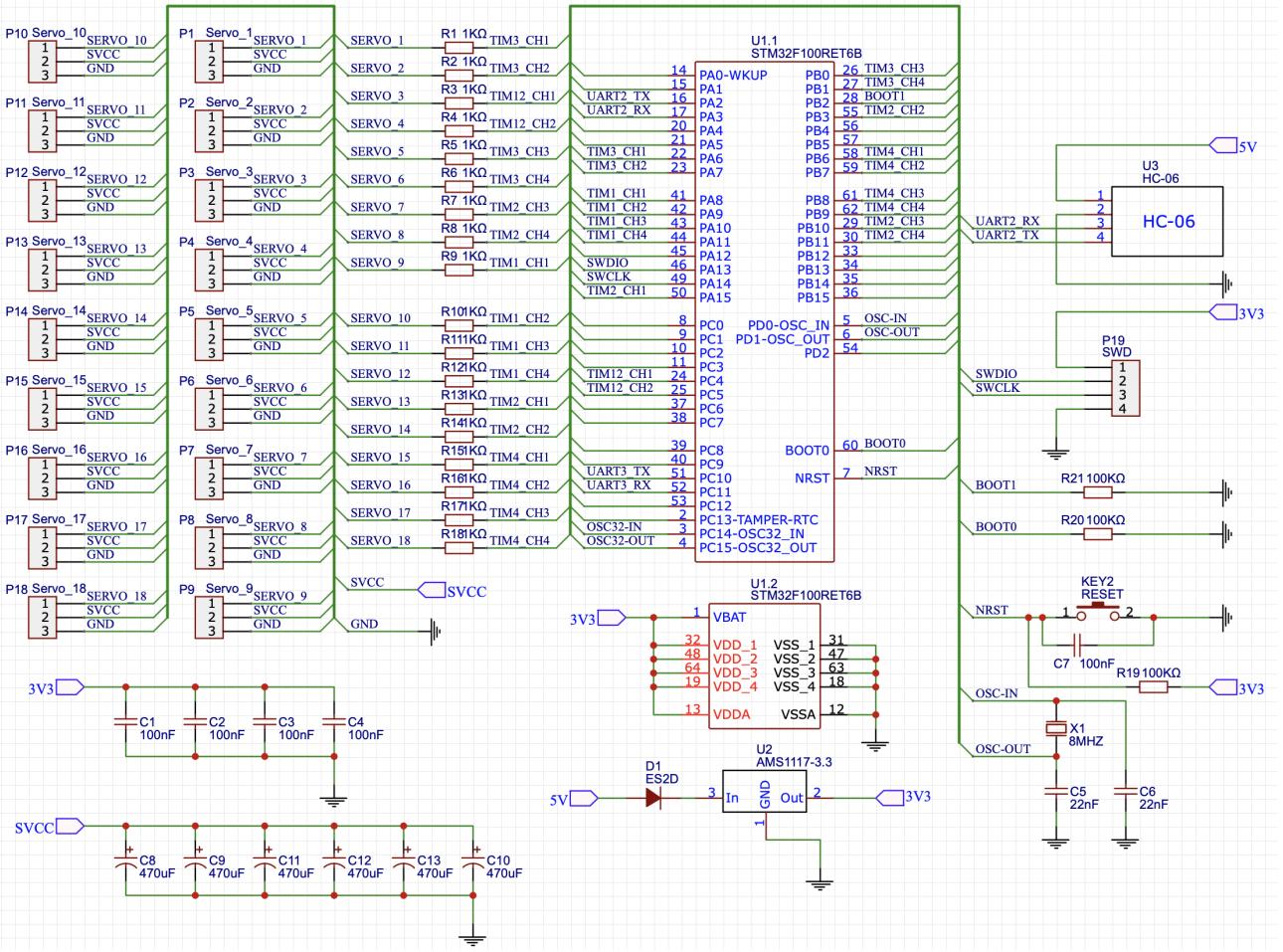


Рисунок 7 – Принципиальная схема платы управления гексаподом

Таким образом, для управления гексаподом применяется статический баланс, при котором, как описывалось ранее, проекция центра масс робота находится в пределах периметра, образованного точками опоры. Генерация алгоритма движения суставов ног производится за помощью пакета ФРУНД, путем решения обратной задачи динамики. В роли исходных параметров выступают траектории движения концов ног робота, и его центра масс.

Для передачи сгенерированных параметров непосредственно в робота используется интерфейс UART – универсальный асинхронный приемник / передатчик – это периферийное устройство микроконтроллера, которое преобразует входящие и исходящие байты данных в последовательный битовый поток.

Передача по usart осуществляется следующим образом, как показано на рисунке 8:

- В состоянии ожидания вывод трансмиттера находится в положении логической единицы;
- Передача байта начинается со стартового бита, который является

логическим нулем;

- Передача байта оканчивается битом, который является логической единицей;
- Передача осуществляется с младшего бита;
- В конце может передаваться бит четности (опционально).

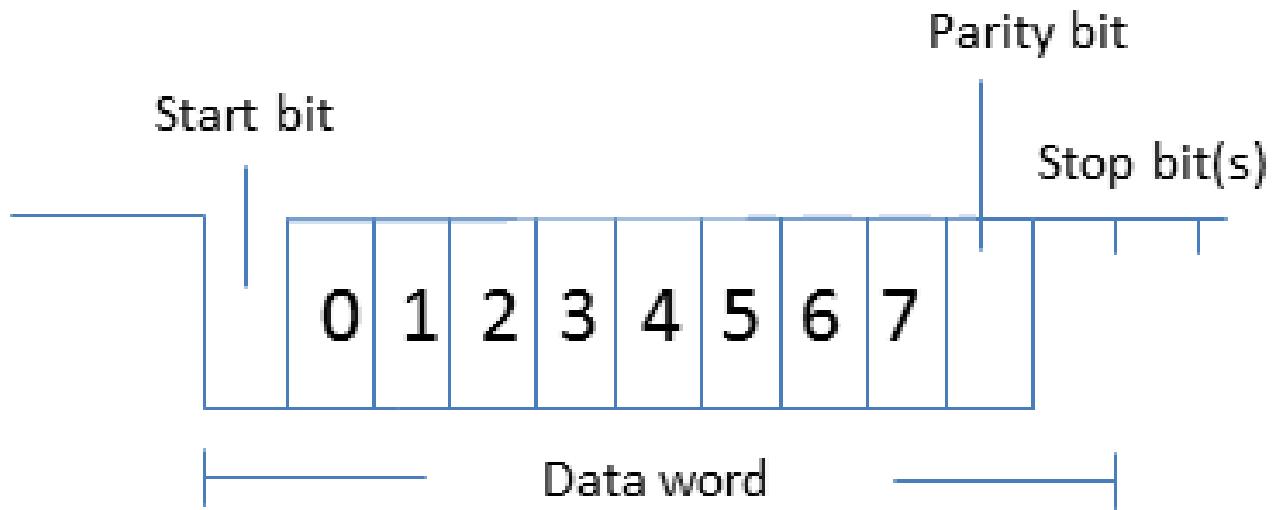


Рисунок 8 – UART Serial Data

## 2.2 Общая структура системы управления

На рисунке 9 показана общая структура управления гексаподом

Оператор задает необходимые параметры для движения и требуемое направление походки в устройстве управления, которое, в свою очередь, формирует их определенным способом для передачи в генератор походки, т.е. ФРУНД посредством использования сокетов.

Сокеты обеспечивают связь между двумя разными процессами на одной и той же машине. Точнее, это способ общения с другими компьютерами с использованием стандартных файловых дескрипторов Unix. В Unix каждое действие ввода / вывода выполняется путем записи или чтения файлового дескриптора. Дескриптор файла - это просто целое число, связанное с открытым файлом, и это может быть сетевое соединение, текстовый файл, терминал или что-то еще.

Генератор походки в данном случае является основным компонентом в системе. Получая данные от устройства управления, выполняется расчет движений для текущего момента времени и управляющий алгоритм для всех

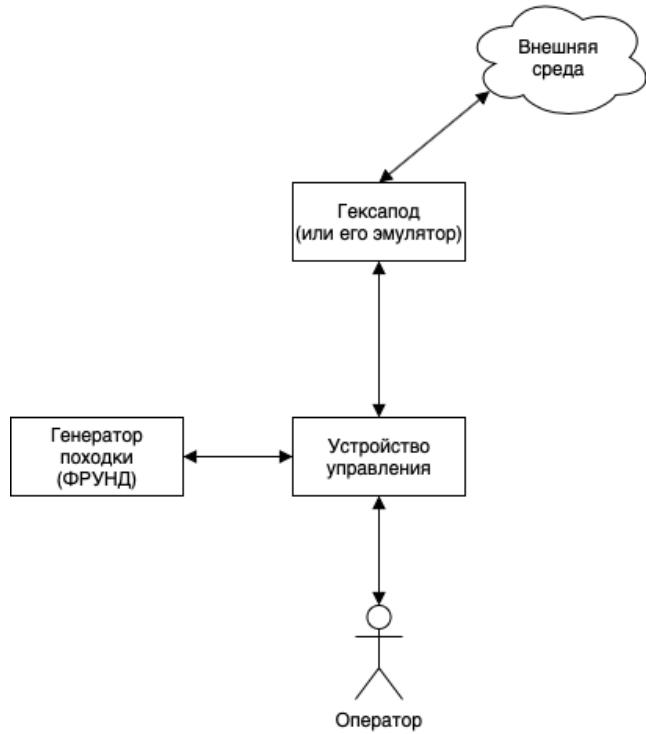


Рисунок 9 – Структура системы управления

суставов робота, которая отправляется непосредственно в гексапод или его эмулятор.

### 2.3 Проектирование системы управления роботом-гексаподом

Для реализации системы управления необходимо выбрать устройство, благодаря которому будет осуществляться управление. Для начала необходимо выделить необходимые управляющие параметры и их количество. А затем под существующие потребности выбрать наилучший способ реализации интерфейса управления для оператора.

#### 2.3.1 Спецификация команд управления

Реализация команд управления реализована в текстовом пакетном режиме – команды записываются в виде строки вещественных чисел. Пример содержимого в файле сокета команд представлен в таблице 1:

Значения параметров команд:

- 1) Time stamp;
- 2) Величина изменения скорости заданной точки робота в направлении X (условные единицы, значение 1 увеличение (уменьшение -1) ско-

Таблица 1 – Пример содержимого файла команд

TS	DX	DY	DZ	DUX	DUY	DUZ	LST	HST	SM	FB	GT
0.5	5	0	-3	0	0	0	0	0	0	0	0
2.05	0	-3	0	0	1	0	0	0	0	0	0
3.15	0	-3	0	0	0	0	0	0	0	0	2
3.8	0	-3	0	1	0	0	0	0	0	0	0
19.2	-2	-1	-3	0	0	0	0	0	0	0	0
31.4	0	0	0	0	0	0	0	0	0	0	0

ности на фиксированную величину с фиксированным ускорением, модуль скорости ограничен);

- 3) Величина изменения скорости заданной точки робота в направлении Y (условные единицы, аналогичные п. 2);
- 4) Величина изменения вертикальной координаты заданной точки робота в направлении Z (условные единицы, значение 1 увеличение (уменьшение -1) высоты на фиксированную величину за фиксированное время, максимальные и минимальные значения ограничены);
- 5) Величина изменения угла относительно продольной оси X робота (условные единицы, значение 1 увеличение (уменьшение -1) угла на фиксированную величину за фиксированное время, максимальные и минимальные значения ограничены);
- 6) Величина изменения угла относительно поперечной оси Y робота (условные единицы, аналогично п. 5);
- 7) Величина изменения угловой скорости робота относительно вертикальной оси Z (условные единицы, значение 1 увеличение (уменьшение -1) скорости на фиксированную величину с фиксированным ускорением, модуль скорости ограничен);
- 8) Параметр длины шага - условные единицы, значение 1 увеличение (уменьшение -1) длины шага на фиксированную величину, максимальные и минимальные значения ограничены);
- 9) Параметр высоты шага - условные единицы, аналогично п. 8;
- 10) Параметр походки — соотношение времени переброса ко времени опоры, аналогично п. 8;
- 11) Параметр походки — поперечное расстояние от стопы до кузова, аналогично п. 8;

12) параметр походки — тип походки — целые числа 1, 2, 3,....

### 2.3.2 Постановка и анализ требований

В существующей спецификации команд можно выделить главные и второстепенные параметры. Главные параметры нуждаются в постоянном контроле со стороны оператора, т.к. они отвечают за непосредственное перемещение гексапода в пространстве. В данном случае главными будут являться следующие параметры:

1) Линейные скорости:

- DX – Величина изменения скорости по оси X;
- DY – Величина изменения скорости по оси Y;
- DZ – Величина изменения скорости по оси Z.

2) Угловые скорости:

- DUX – Величина изменения скорости вращения относительно оси X;
- DUY – Величина изменения скорости вращения относительно оси Y;
- DUZ – Величина изменения скорости вращения относительно оси Z.

Таким образом, к аппаратному устройству управления выделяются следующие требования: пульт должен иметь возможность удобной подстройки всех главных параметров, необходимых для управления гексаподом.

Все прочие параметры, а именно:

- Параметр длины шага,
- Параметр высоты шага,
- Параметр отношения времени переброса ко времени опоры,
- Параметр поперечного расстояния от стопы до кузова,
- Параметр типа походки,

названные выше второстепенными, безусловно, тоже являются важными. Но они не нуждаются в постоянном изменении в процессе ходьбы, поэтому в их присутствии на аппаратном пульте нет никакой необходимости.

Следующие требования предъявляются к виртуальному пульту управления:

- Возможность управления всеми параметрами;
- Интуитивный интерфейс управления;
- Адаптивность интерфейса управления;
- Возможность использования данного интерфейса на большинстве современных платформ;
- Оптимизация интерфейса под устройства с сенсорным экраном;
- Синхронизация заданных параметров между устройствами, на которых открыт интерфейс управления.

### 2.3.3 Проектирование архитектуры системы управления

В результате проектирования архитектуры системы управления для данного робота-гексапода был получен общий подход, который можно использовать для улучшения данной системы в целом или для построения других аналогичных шагающих роботов и систем управления для них. Разработанная архитектура системы управления изображена на рисунке 10.

При осуществлении управления путем использования "Hardware joystick" данные со стиков джойстика передаются по радиочастотам в "Joystick receiver". Далее через USB-USART преобразователь по протоколу IBUS информация о текущем положении стиков джойстика передается в "Joystick receiver bridge". Функциями данного модуля являются прием и обработка параметров. Обработка в данном случае заключается в том, чтобы по определенному алгоритму преобразовать сырье данные, полученные от пульта, к формату интерфейса "Walking params", который ожидает "FRUND bridge".

Если же управление осуществляется через "Web control interface" параметры, путем использования http запросов попадают непосредственно на HTTP сервер. Если запущено несколько экземпляров "Web control interface" на одном или нескольких устройствах, данные между ними синхронизируются через websocket, дабы избежать той ситуации, что каждый экземпляр будет иметь и передавать свои локальные параметры, которые могут сильно отличаться от других. Аналогично управлению с "Hardware joystick" в конечном итоге данные также преобразуются для передачи через интерфейс "Walking params" во "FRUND bridge".

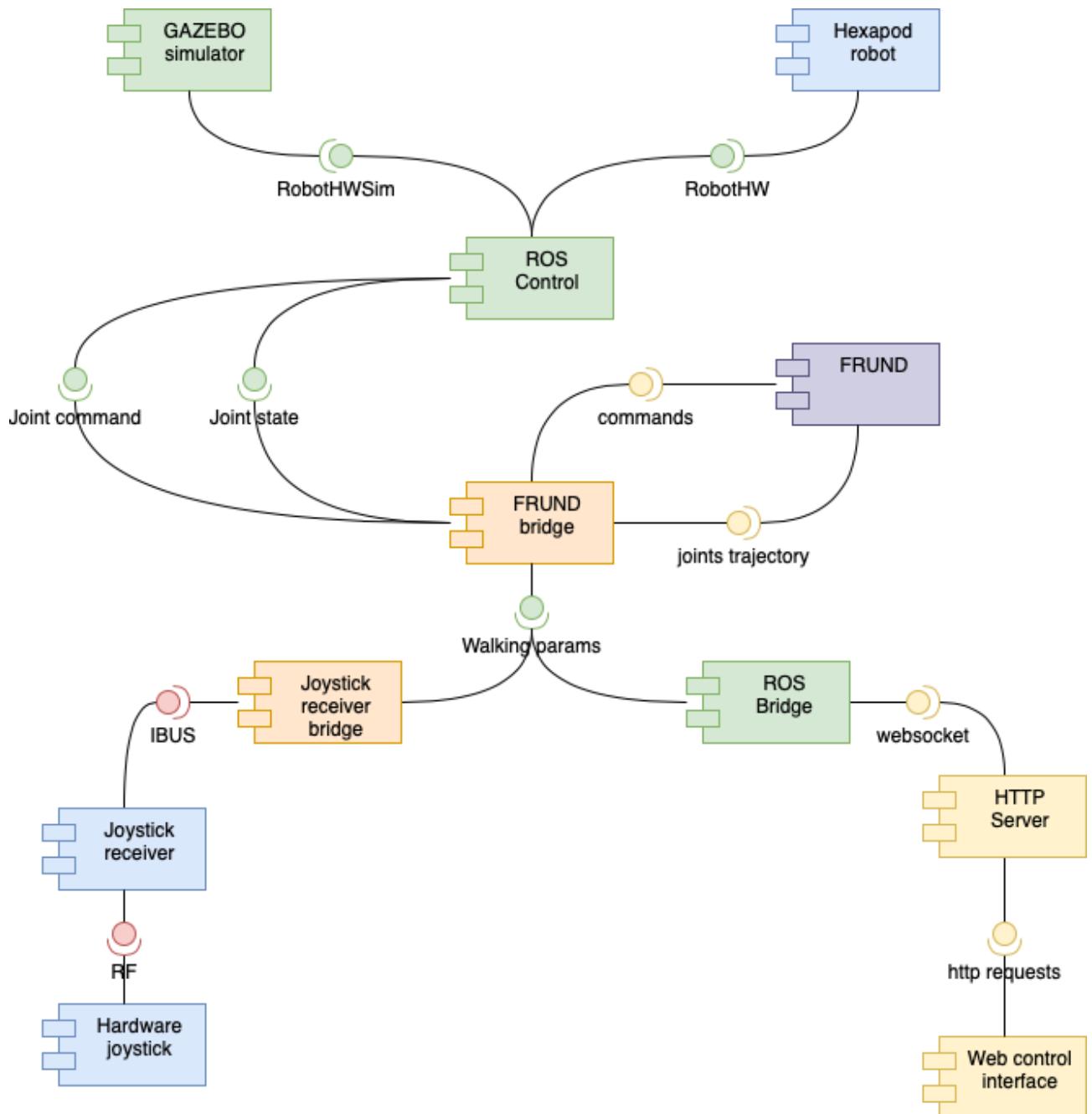


Рисунок 10 – Архитектура системы управления роботом-гексаподом

Отдельно стоит выделить модуль "FRUND bridge", который преобразует принятые "Walking params" к в специальную последовательность, которую ожидает ФРУНД, а затем посредством сокетов через интерфейс "commands" передает в него сформированные команды управления. Далее по составленной математической модели робота-гексапода (рис. 11) ФРУНД генерирует траектории перемещения суставов и также посредством сокетов, используя интерфейс "Joints trajectory" передает их обратно во "FRUND bridge".

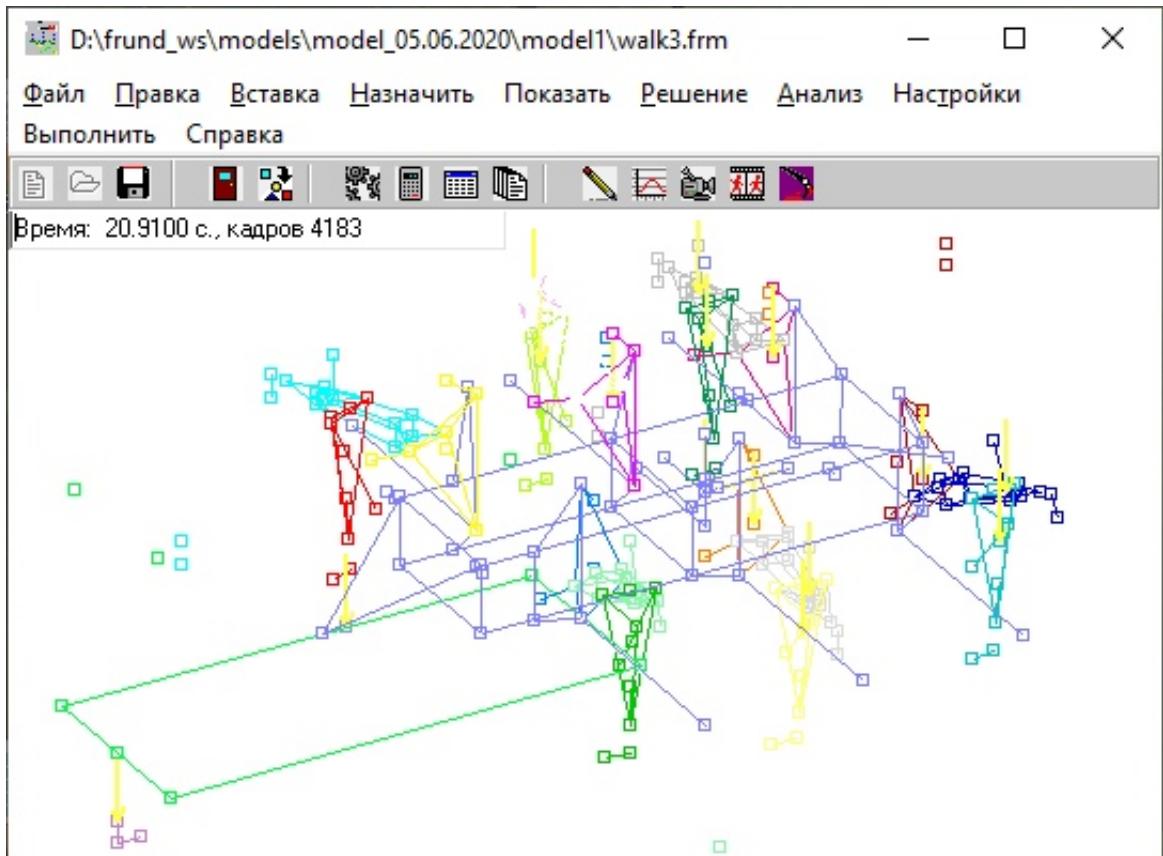


Рисунок 11 – Модель робота-гексапода в системе ФРУНД

Следующим модулем системы является "ROS control", который выделяет следующие интерфейсы для входных параметров: "Joint command" и "Joint state". Соответственно "FRUND bridge" принимая сгенерированные во ФРУНД траектории должен преобразовать их к определенному формату и передать их, используя данные интерфейсы.

Далее следует довольно стандартная концепция использования пакета "ROS control" в связке с GAZEBO и/или аппаратной моделью робота. Обратившись к мануалу можно увидеть следующий обзорный пример (рисунок 12) взаимосвязи между симуляцией, оборудованием, контроллерами и передачами.

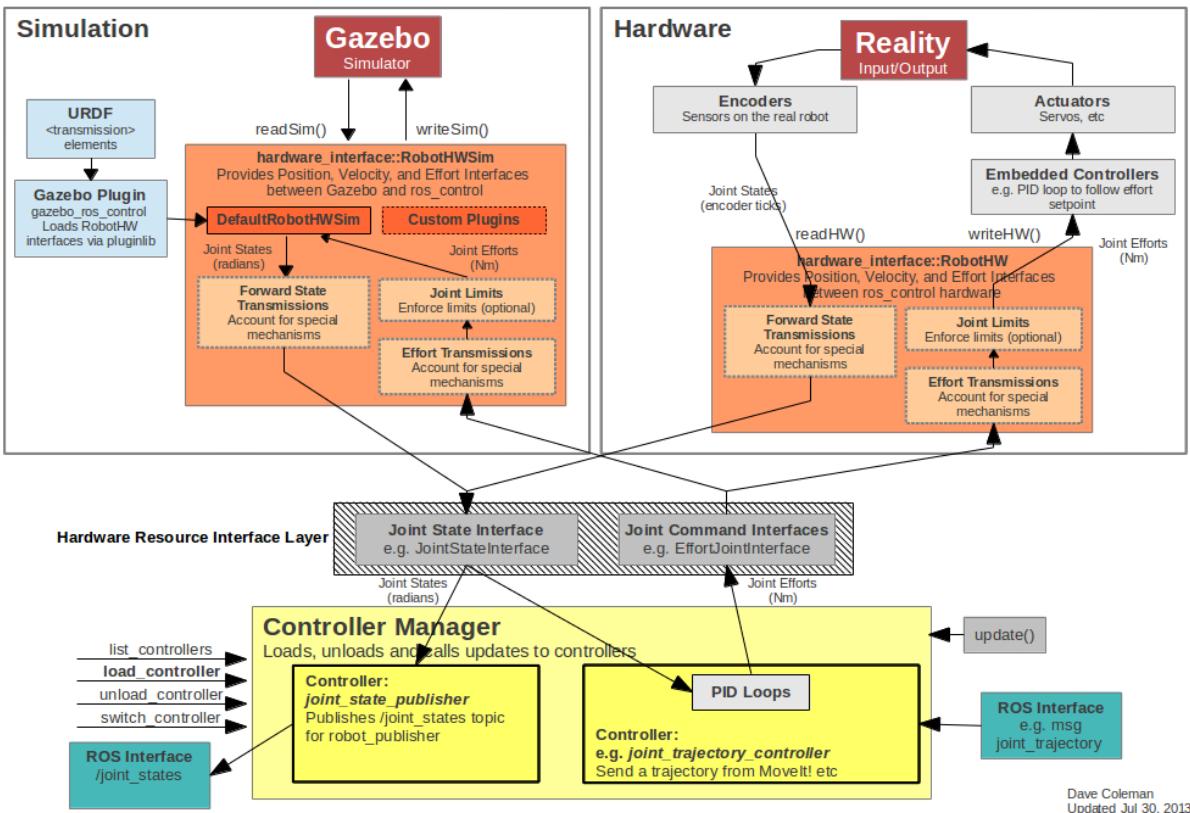


Рисунок 12 – Пример использования связки GAZEBO + ROS + ros\_control

Передача – это элемент в конвейере управления, который трансформирует переменные усилия / потока так, что их продукт – мощность – остается постоянным. Реализация интерфейса передачи отображает переменные усилия / потока на выходные переменные усилия / потока при сохранении мощности.

Таким образом, по умолчанию `gazebo_ros_control` очень прост, хотя его также можно расширить с помощью дополнительной архитектуры плагинов, которая позволяет создавать собственные пользовательские интерфейсы аппаратного обеспечения робота между `ros_control` и Gazebo.

## 2.4 Обзор решения-аналога

После изучения некоторого количества публикаций на схожую тематику, была найдена статья "Web-Based Robot Control Interface" изданная в журнале "IOP Conference Series: Earth and Environmental Science". В ней авторы рассказывают о создании системы управления для Robotic hand.

Их архитектура системы состоит из пяти модулей. Серверная часть включает в себя четыре модуля: модуль входа в систему для регистрации пользователя, модуль отображения и управления трехмерной имитационной моделью, модуль последовательной передачи данных, робот и однокристальный модуль. Сторона браузера содержит один модуль: модуль веб-интерфейса управления роботом. Структура показана на рисунке 13.

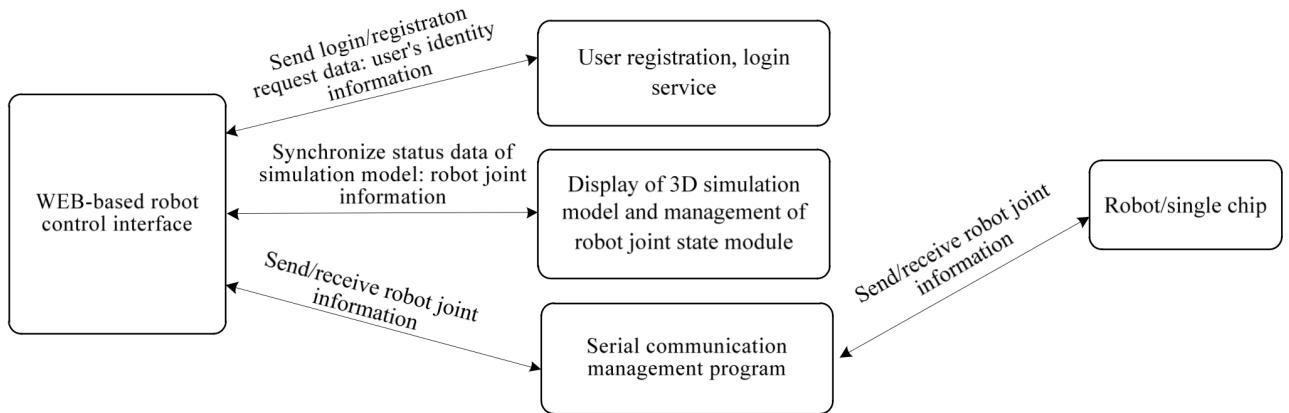


Рисунок 13 – Структура решения–аналога

#### 2.4.1 Архитектура и сценарий использования

В работе предполагается следующий сценарий использования: пользователь открывает браузер через компьютер или мобильное устройство. Веб-интерфейс управления роботом открывается при вводе URL-адреса и автоматически переносится в интерфейс входа пользователя. Интерфейс входа в систему имеет ссылку для регистрации, через которую пользователь может получить доступ к странице регистрации. Интерфейс входа и регистрации основан на сервисе PHP5. Поскольку PHP работает на стороне сервера, клиент не может знать конкретную логику кода службы регистрации / входа в систему, который обеспечивает безопасность идентификационной информации. После подтверждения заявки на регистрацию пользователь может войти в систему с именем пользователя и паролем. После получения идентификационной информации пользователя о входе в систему служба входа в систему считывает файл JSON разрешенного белого списка пользователей и проверяет, существует ли полученная пара имени пользователя и пароля в белом списке. Если это так, выполняется переход на веб-интерфейс управления роботом.

Веб-интерфейс управления роботом состоит из JS-модуля отображения имитационной модели, JS-модуля управления соединением на основе ползунка и JS-модуля отображения информации о последовательном порту. Технология WebGL используется для отображения трехмерной модели робота имитационной модели на странице и отслеживания состояния соединения робота. Имитационная модель обновляется в соответствии с изменениями состояния соединения робота. Модуль управления соединением на основе ползунка считывает модель робота, извлекает все соединения робота и создает взаимно-однозначное соответствие между ползунком и соединением. Модуль отображения информации о последовательном порте подписывается на тему данных о последовательном порте. Когда новое сообщение выпущено, модуль анализирует содержимое сообщения и обновляет информацию данных соединений в веб-интерфейсе управления. Когда пользователь перемещает ползунок в интерфейсе управления для генерации действия, код JS считывает новую информацию о соединении и публикует сообщение в теме состояния соединения. После получения сообщения программа имитационной модели ROS обновляет объединенную информацию имитационной модели и отправляет данные в веб-интерфейс, так что имитационная модель веб-интерфейса также изменяется.

ROS-узел управления последовательной связью также подписывается на тему совместного состояния. При получении новой информации о совместном состоянии информация анализируется, и вызывается библиотека Rosserial для записи данных в последовательный порт. Пакет данных последовательно отправляется с помощью кабеля последовательной передачи данных в микроконтроллер, который управляет роботом. Микроконтроллер имеет собственную управляющую логику, позволяющую менять узлы робота, реализовывать дистанционное управление роботом через веб-интерфейс. В то же время узел управления последовательной связью опубликует сообщение о состоянии соединения в разделе с информацией о данных последовательного порта при чтении совместной информации об оборудовании робота из кабеля данных последовательного порта. Модуль JS с информацией о последовательном порте в клиентском веб-интерфейсе управления считывает и анализирует содержимое сообщения. Соединения и значения обновляются в веб-интерфейсе для реализации функции отображения реальной информации

ции о соединениях на стороне робота на клиентском интерфейсе.

Полная архитектура проекта управления роботизированной рукой отражена на рисунке 14.

#### 2.4.2 Анализ и сравнение с предложенным решением

Решение коллег, безусловно, является очень ценным, хоть и не относится к управлению роботом-гексаподом. Однако принцип который был предложен ими для создания архитектуры системы управления роботизированной рукой, является довольно схожим.

Как можно было понять из описания выше, и скриншоту готовой панели управлении, изображенной на рисунке 15, а также по фото совместно с устройством, изображенном на рисунке 16, веб интерфейс путем использования технологии WebGL отображает все изменения которые должны происходить в ходе обработки команд на имитационной модели в режиме реального времени. Данная опция является очень интересной в визуальном плане и удобной в плане отладки устройства и поиска возможных технических неисправностей, сравнивая движения, выполняемые идеальной виртуальной имитационной и реальной моделями. Поэтому было решено добавить данную возможность в следующих версиях системы для робота-гексапода.

Однако возможность регистрации пользователей и белый список, являются излишними, по крайней мере в данной выпускной работе, потому как аппаратура и ПО не находится в общем доступе. Управление устройством сторонними людьми, не обученными и не компетентными в данной сфере не допускается. Все публичные демонстрационные тесты должны проводиться строго под наблюдением специалистов.

#### 2.5 Выводы по главе

Была разработана общая структура, а также архитектура системы управления роботом-гексаподом с использованием системы ФРУНД, как основного компонента генерации движений. Изучена стандартная архитектура построения систем с использованием пакета ros\_control в связке с ROS и GAZEBO.

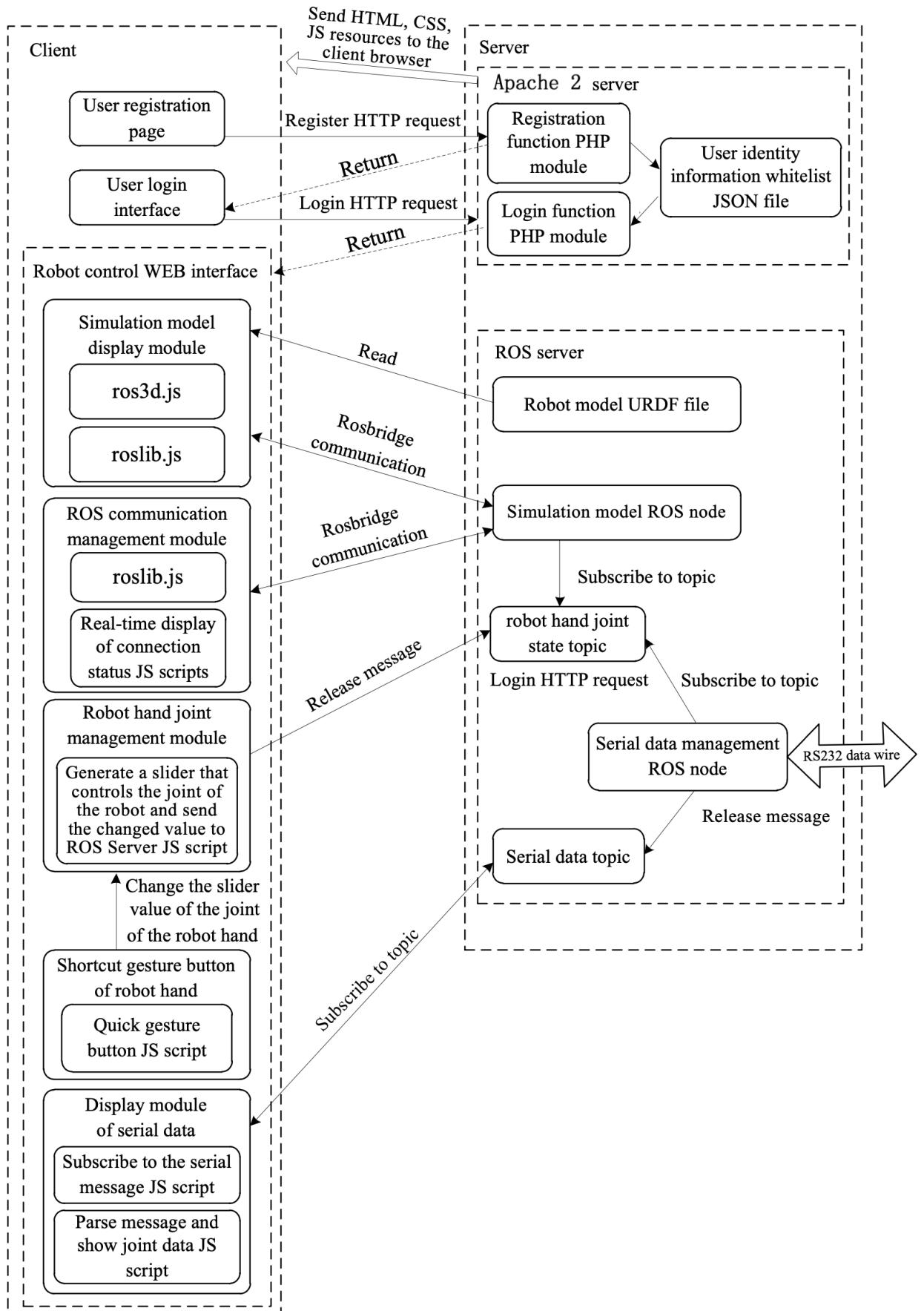


Рисунок 14 – Архитектура системы программного обеспечения  
решения–аналога



Рисунок 15 – Robotic hand control UI

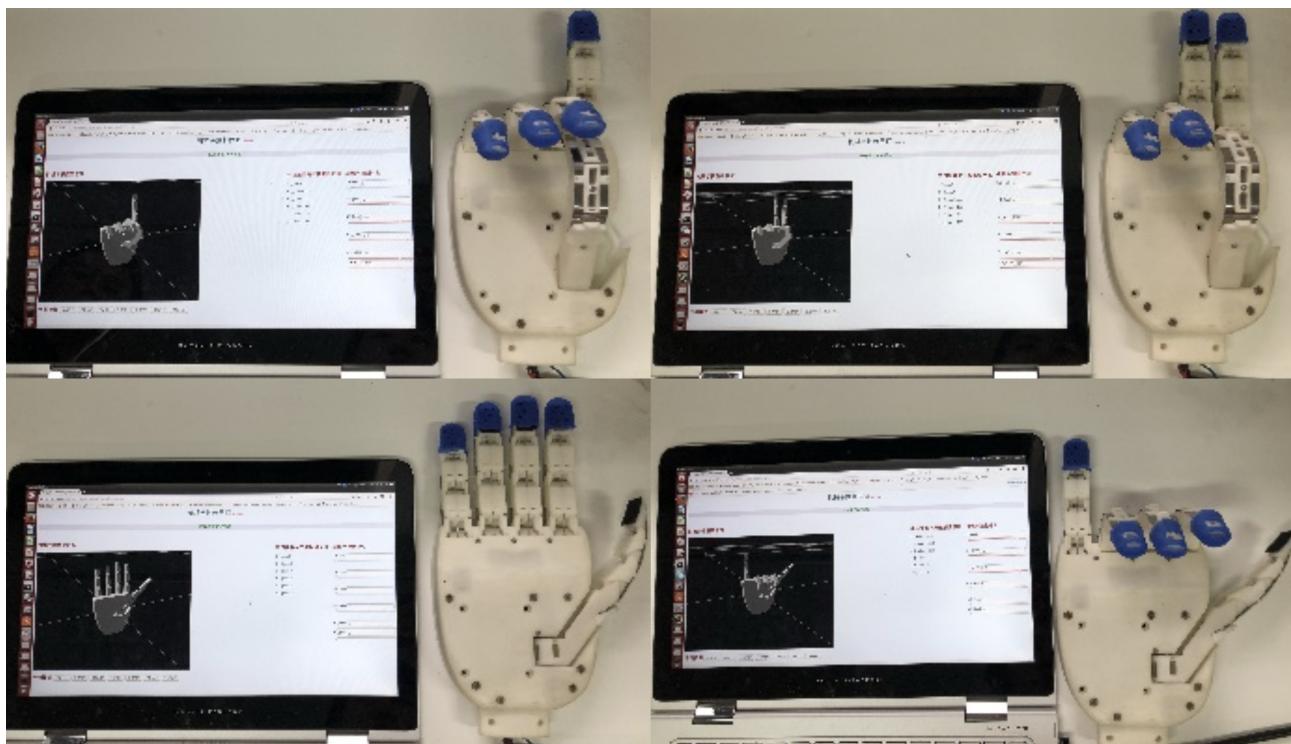


Рисунок 16 – Демонстрация работы системы управления роботизированной рукой

Выделены основные критерии и требования разработки системы управления. Рассмотрены плюсы и минусы решения-аналога, которые будут учтены для последующей модернизации решения представленного в третьей главе.

Предложенная концепция использует ROS как связующий слой модулей системы и позволит использовать данное решение для управления новыми моделями робота, а также для модернизации и расширения функционала

текущей модели, путем добавления новых программных и аппаратных модулей.

## 3 РЕАЛИЗАЦИЯ СИСТЕМЫ УПРАВЛЕНИЯ РОБОТОМ ГЕКСАПОДОМ

В этой главе описывается этап реализации системы управления роботом-гексаподом на основе спроектированной во второй главе архитектуры. Рассматриваются необходимые компоненты, их связь друг с другом, а также принцип работы и примеры.

### 3.1 Sketch

Для проектирования и визуализации UI (User Interface) веб версии пульта управления роботом был выбран Sketch – векторный графический редактор для macOS, разработанный голландской компанией Bohemian Coding, предназначенный исключительно для создания интерактивных прототипов веб-приложений и дизайнов приложений. Основные преимущества и недостатки данного редактора отражены в таблице 2 В отличие от Photoshop, графическое программное обеспечение Sketch изначально создавалось для разработки интерактивных интерфейсов. Он создает живые композиции, чтобы была возможность просматривать, что происходит, когда пользователь щелкает, смахивает или касается нарисованного дизайна на своем настольном компьютере, ноутбуке, планшете или телефоне.

Проектирование дизайна веб-пульта для робота-гексапода изображено на рисунке 17.

Таблица 2 – Преимущества и недостатки Sketch

Преимущества	Недостатки
Интуитивный интерфейс	Только MacOS
Быстрое обучение	Только одно устройство на одну лицензию
Полезные функции для совместной работы	Нет автоматической разметки
Работает с активами Illustrator	



Рисунок 17 – Проектирование Web UI пульта управления в Sketch

### 3.2 Front-end и VueJS

Для реализации веб интерфейса управления был выбран Vue – современный javascript фреймворк. На первый взгляд, можно сказать, что библиотека Vue - это просто смесь Angular и React. Фактически, Evan You, создатель Vue, заимствовал концепции из Angular и React. Например, концепция использования Vue предполагает сохранение логики и макетов компонентов вместе с таблицами стилей в одном файле. Это похоже на то, как React работает без таблиц стилей. Чтобы позволить компонентам Vue общаться друг с другом, Vue использует реквизиты и объекты состояния. Этот подход также существовал в React до того, как Vue приняла его.

Как и в Angular, во Vue предполагается смешивание макетов HTML с JavaScript. При верстке необходимо использовать директивы Vue, такие как v-bind или v-if, чтобы интерполировать значения из логики компонента в шаблоны.

Одна из причин, по которой Vue стоит рассматривать вместо React, заключается в том, что библиотека Redux часто используется в крупномасштабных приложениях React. Как объясняется в разделе React, когда приложение React + Redux становится больше, требуется тратить много времени на внесение небольших изменений в несколько файлов вместо того, чтобы

фактически работать над функциями. Библиотека Vuex - Flux-подобный инструмент управления состоянием, разработанный для Vue кажется менее громоздким, чем Redux.

Что касается выбора между Vue и Angular, причины выбора Vue вместо Angular могут быть сведены к следующему: Angular - это слишком сложная, полноценная структура с ограничительным характером; Vue намного проще и менее строг, чем Angular.

Также стоит отметить, что для использования Vue по сравнению с Angular и React, не требуется изучать ничего кроме JavaScript. Vue не требует другой расширенный набор JavaScript, такой как TypeScript (для Angular) или JSX (для React). В результате обучение работе с Vue быстрее, чем с Angular и React.

Одним из недостатков Vue является то, что он все еще гораздо менее популярен, чем React или Angular. Хорошим показателем использования инфраструктуры JS является количество вопросов, возникающих на Stack Overflow. Вопросы показывают, с какими фреймворками чаще всего сталкиваются разработчики, хотя это также говорит об их уровне сложности.

В этом случае впервые в истории вопросы в 2019 году по React слегка опередили вопросы по Angular. Между тем, вопросы Vue имеют тенденцию к росту, но все еще остаются низкими. Статистика по вопросам использования Angular.js (старой версии) упала, как и ожидалось. Статистика вопросов по Javascript front-end фреймворкам изображена на рисунке 18. Таким образом, у Vue гораздо меньше готовых решений, поэтому при решении нетривиальных задач приходится тратить немного больше времени на разработку.

Экосистема VueJS состоит из:

- Vue как библиотека представления;
- Vue-загрузчик для компонентов;
- Vuex, выделенная библиотека для управления состоянием приложения с помощью Vue, Vuex близок по концепции к Flux;
- Vue.js devtools для Chrome и Firefox.
- Nuxt.js, проект, предназначенный для создания серверных приложений с помощью Vue. Nuxt.js является конкурентом для Angular Universal;
- Weex, библиотека JS, которая поддерживает синтаксис Vue и ис-

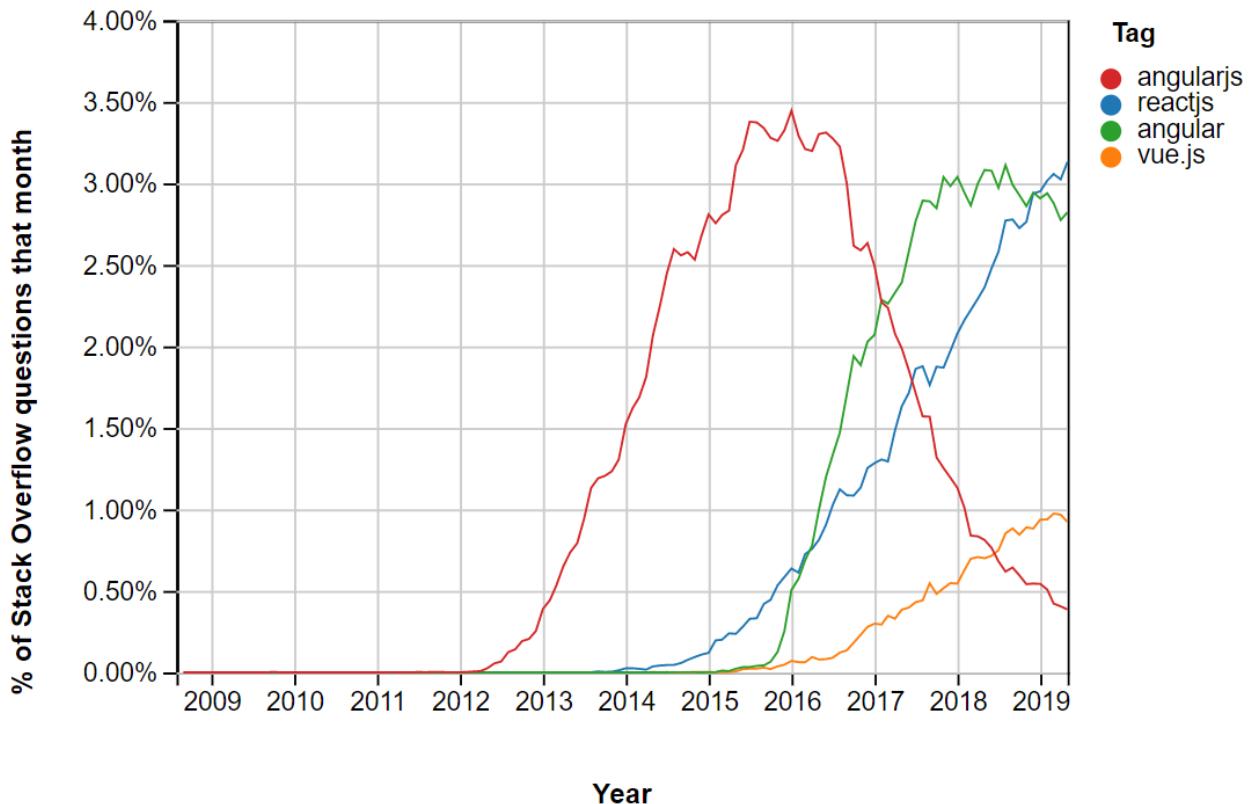


Рисунок 18 – Статистика вопросов по JS front-end фреймворкам на Stack Overflow

пользуется для мобильной разработки.

### 3.3 Browser-Sync

BrowserSync - это инструмент автоматизации, который ускоряет веб-разработку, настройку и тестирование за счет синхронизации изменений файлов и взаимодействия на многих устройствах.

Особенности BrowserSync:

- Перезагрузка в реальном времени – это, вероятно, самая важная функция BrowserSync. При изменении кода страница автоматически перезагружается. Прямая перезагрузка работает во многих браузерах и устройствах;
- Синхронизация взаимодействия – это означает, что все действия отражаются во всех браузерах. Эта небольшая функция полезна для тестирования, особенно при тестировании на многих устройствах. Также можно настроить, какие действия отражаются в разных браузерах;
- Имитация более медленных соединений – используется для тести-

рования при интернет-соединении, благодаря тому, что BrowserSync имеет функцию, которую можно использовать для регулирования скорости соединения сайта;

- История URL – BrowserSync регистрирует всю историю просмотров, чтобы можно было отправить тестовый URL на все устройства;
- BrowserSync совместим со многими исполнителями задач, такими как GULP и Grunt. И они работают во многих операционных системах.

Принцип работы заключается в том, что BrowserSync создает небольшой сервер, но если уже имеется настройка сервера, BrowserSync может подключиться к этому серверу и действовать как прокси. Затем он вставляет файл javascript на каждую страницу. Этот файл использует WebSockets для связи между сервером и клиентом для отслеживания изменений в коде или действиях браузера. Как только BrowserSync обнаруживает действие, он выполняет перезагрузку страницы.

### 3.4 ROS

ROS (Robot Operating System) предоставляет стандарт для разработки программного обеспечения для робототехники, который можно использовать с любым роботом. Этот стандарт позволяет фактически сосредоточиться на ключевых функциях приложения, используя существующую основу, таким образом реализуя middleware слой.

ROS в основном разработан с использованием 2 языков: C++ и Python. Это часто наиболее предпочтительные и используемые языки при разработке приложений для робототехники. Библиотека roscpp используется для написания кода на C++, а библиотека rospy для написания кода на Python. Есть также несколько библиотек для создания моста с другими языками, такими как rosjava для Java и roslibjs или rosnodejs для JavaScript.

С помощью ROS можно легко разделить кодовую базу на пакеты, содержащие небольшие программы, называемые узлами. Для передачи данных между этими узлами ROS содержит 3 основных средства коммуникации:

- Топики используются в основном для отправки потоков данных между узлами. К примеру, отслеживаются некоторые параметры на роботе. Узел, контролирующий эти параметры, отправит поток дан-

ных с этими параметрами. Теперь любой другой узел может подпisyаться на этот топик и получить данные (рис. 19);

- Сервисы позволяют создать простую синхронную клиент-серверную связь между узлами. Используется для изменения настроек робота или для запроса определенного действия: включить режим походки, запросить определенные данные и так далее;
- Действия основаны на топиках. Они предназначены для того, чтобы предоставить асинхронную архитектуру клиент / сервер, где клиент может отправить запрос, который занимает много времени (например, запросить перемещение робота в другое место). Клиент может асинхронно отслеживать состояние сервера и отменять запрос в любое время.

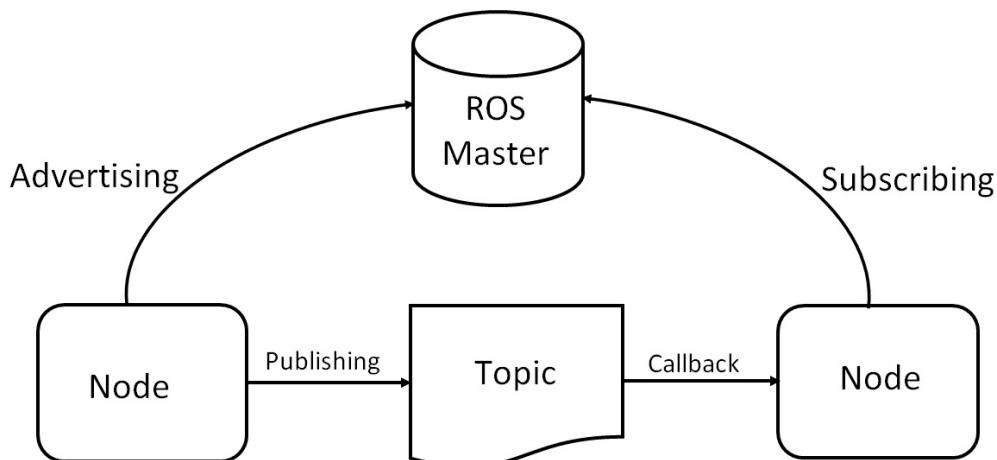


Рисунок 19 – Топик как способ коммуникации между двумя и более узлами

При создании топика необходимо указать тип сообщений (messageType) который будет передаваться с его помощью. В ROS существуют встроенные типы сообщений, однако этого не всегда бывает достаточно, поэтому имеется возможность создания кастомных сообщений по необходимости. В данной работе дополнительно был создан FrundControl, для передачи параметров управления от пульта, листинг приведен ниже.

```
# Hexapod walking parameters
# accel           Control movement and rotation
# gait_settings   Extra gait settings

geometry_msgs/Accel accel
hexapod_msgs/GaitParams gait_settings
```

В данном случае accel представляет собой стандартный тип сообщений, содержащий в себе ускорение в свободном пространстве, разбитое на его линейную и угловую части:

```
# This expresses acceleration in free space broken into
# its linear and angular parts.

Vector3 linear
Vector3 angular
```

где linear и angular представляет собой:

```
float64 x
float64 y
float64 z
```

Gait\_settings также является кастомным сообщением, входящим в состав FrundControl и содержит в себе дополнительные настраиваемые параметры походки:

```
# Extra gait settings for FRUND gait generator

float32 step_length
float32 step_height
float32 support_movement
float32 foot_body_ratio
uint8   gait_type
```

Стоит отметить, что ROS не зависит от языка. Написанные подпрограммы (узлы) могут быть написаны на любом языке. Таким образом, приложение может иметь узел, написанный на Python, взаимодействующий с узлом, написанным на C++ (рис. 20). Это является возможным, благодаря тому, что уровень коммуникации ниже "языкового уровня". Поскольку он работает как middleware ПО, используются стандартные сокеты TCP / IP для связи между узлами.

ROS не заканчивается набором основных функций и средств коммуникации. Одним из главных преимуществ является возможность использования огромного количества существующих библиотек. Для большинства общих блоков разработки в робототехнике можно найти ROS-пакет с открытым

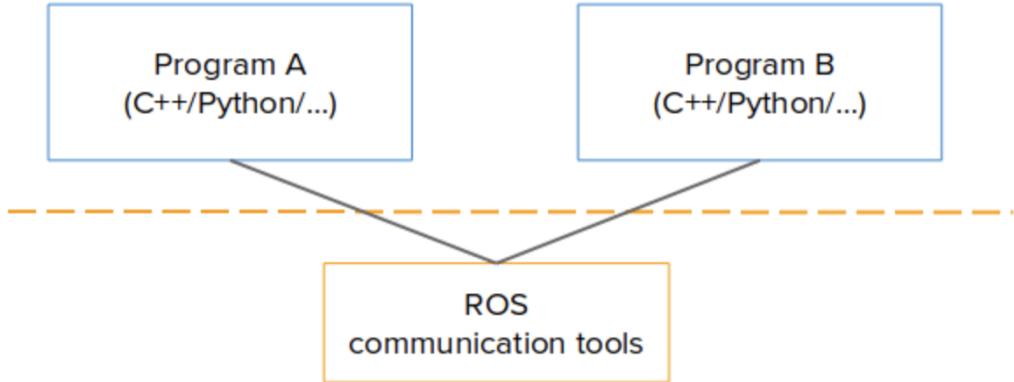


Рисунок 20 – Работа слоя коммуникации ROS

исходным кодом. Пример использования и связи различных компонентов в ROS представлена на рисунке 21.

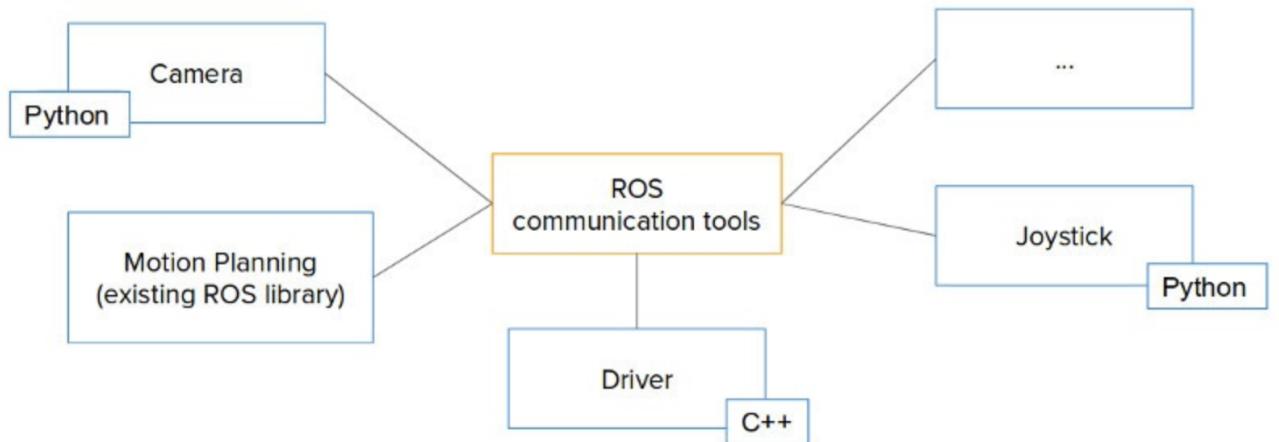


Рисунок 21 – Связь между узлами в ROS

Пример используемых библиотек / пакетов:

- 1) Многие двигатели и камеры имеют существующий драйвер ROS;
- 2) Для основного цикла управления робота используется `ros_control`;
- 3) Чтобы представить трехмерную модель вашего робота, требуется описать ее, используя XML с форматом URDF;
- 4) Для планирования пути и движения используется Moveit. Moveit будет использовать ранее созданный файл URDF;
- 5) Для перемещения мобильного робота вы можете найти полный стек навигации;
- 6) Для 3D визуализации используется Rviz;
- 7) Если требуется более мощный инструмент моделирования с физическими ограничениями, к примеру, с гравитацией и ветром то стоит использовать GAZEBO;

8) Для возможности обмениваться данными между ROS и окружением без ROS используется пакет rosbridge.

Для визуализации графа ROS приложения используется rqt\_graph - это графический плагин из набора инструментов Rqt.

В окне rqt\_graph можно сразу увидеть все работающие узлы, а также связь между ними. Узлы и топики будут отображаться в их пространстве имен. Стоит отметить, что службы ROS в графике rqt не отображаются, только топики (рис. 22). Это связано с тем, что они были реализованы как сервисы.

Рассмотрим компонент FRUND bridge, также реализуемый в рамках реализации данной системы управления. Диаграмма классов данного компонента приведена на рисунке 23.

Диаграмма классов включает в себя два класса, первый из которых (HexapodController) выполняет общие операции взаимодействия с гексаподом и прием данных с пульта. Второй класс (FrundGateway) является шлюзом между ROS топиками и системой ФРУНД. Экземпляр FrundGateway используется в HexapodController для организации передачи управляющих траекторий приводов с ФРУНД в топики, которые сообщают положения joint в ROS control и для передачи управляющих параметров движения (команд) с пульта к системе ФРУНД.

Метод init\_rc\_control выполняет настройку подписчика ros\_subscriber для приема команд из топика, когда приходит очередная порция команд, вызывается функция обратного вызова WalkParamsCallback в которой принятые параметры записываются в буффер для передачи системе ФРУНД. Метод read\_params читает данные (словарь параметров соответствия между joint ROS и каналами во ФРУНД, номер порта, для соединения с ФРУНД по протоколу UDP). Метод FrundExchange представляет собой итерацию обмена данными между ROS и сокетом, который, соединен с системой ФРУНД.

Для обмена данными по сокетам используются два буфера (класс FrundGateway). recv\_buffer принимает массив позиций приводов модели роботов с ФРУНД и затем они читаются в HexapodController с помощью метода ReadBuffer и считанные данные записываются в топики ROS. send\_buffer используется для отправки (записи) параметров движения (команд) и данные

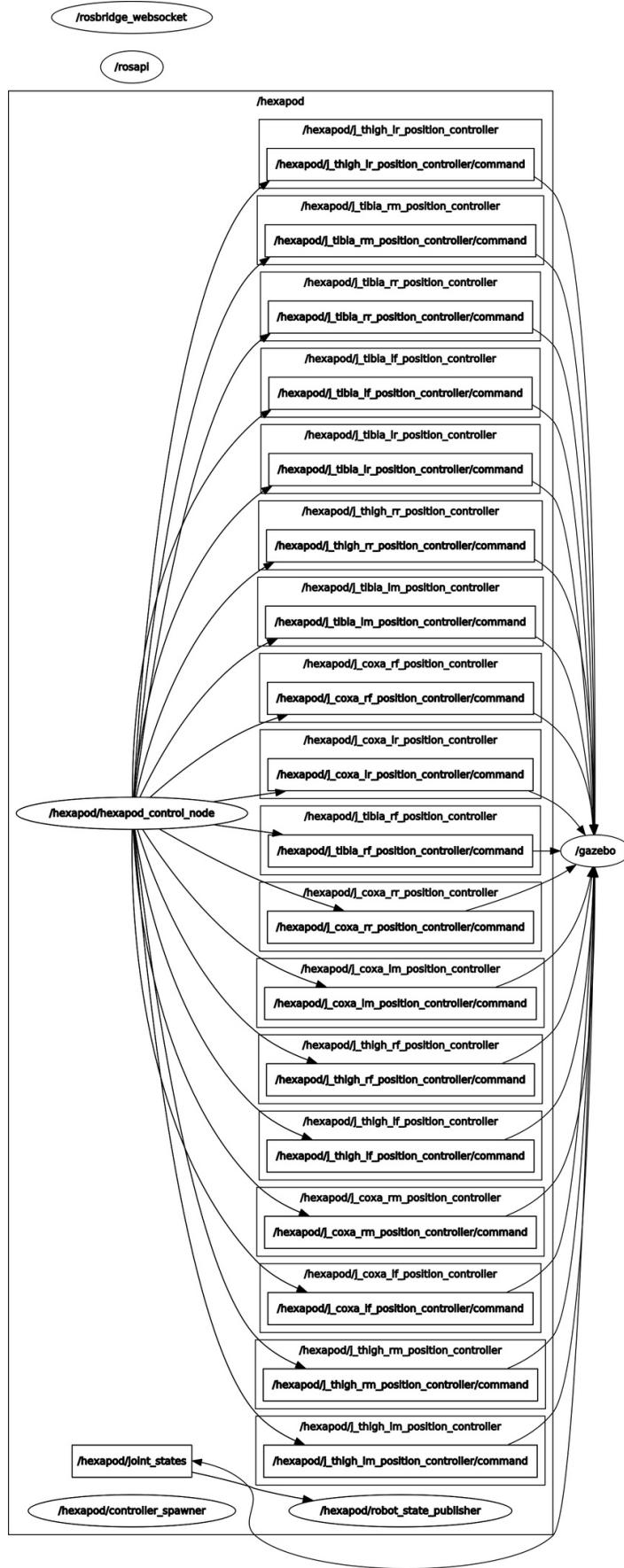


Рисунок 22 – Вывод rqt\_graph для системы управления роботом-гексаподом

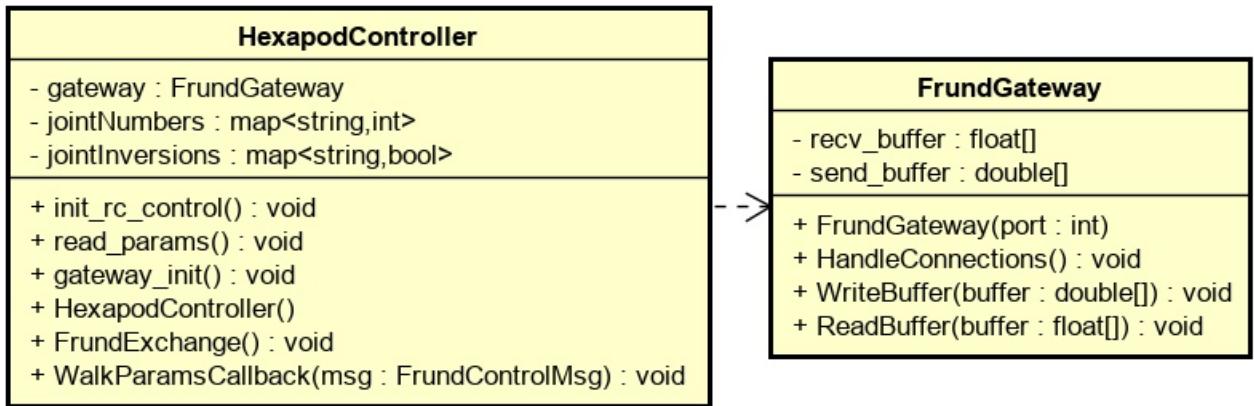


Рисунок 23 – Диаграмма классов компонента FRUND bridge

в него заносятся из HexapodController за помощью метода WriteBuffer.

В конечном итоге взаимодействие между ФРУНД, пультом и моделью робота в ROS организуется путем итераций чтения данных с пульта, записи в буфер отправки, чтения данных с буфера приема, записи их в топики, и обменом буферами между запущенным шлюзом и системой ФРУНД по протоколу UDP по интерфейсу сокетов

### 3.5 Rosbridge

Rosbridge, представляет собой middleware слой абстракции программного обеспечения, предназначенный для разработки приложений, которые напрямую не связаны с роботом. Rosbridge предоставляет простой программный доступ на основе сокетов к интерфейсам и алгоритмам роботов, которые предоставляет ROS. В частности, это облегчает использование веб-технологий, таких как Javascript, с целью расширения использования и полезности роботизированных технологий.

Вычислительные парадигмы с течением времени развивались от автономных настольных систем до клиент-серверных архитектур и от повсеместных веб-приложений. Современные технологии обеспечивают прозрачное администрирование, избыточное хранилище и мгновенное развертывание программного обеспечения, работающего на сильно разнородных платформах, от смартфонов до многоядерных настольных компьютеров.

Поскольку rosbridge был создан для связи ROS и non-ROS приложений существует библиотека rosbridge в Javascript, известная как rosjs. Ее единственная расширенная зависимость - это технология веб-сокетов HTML5. В

настоящее время такие браузеры, как Safari, Opera и Chrome, полностью поддерживают их. Rosjs реализован в виде простой библиотеки Javascript, полностью независимой от предпочтительных сред разработки. Rosbridge построен с использованием сериализованных объектов JSON, которые сами по себе являются базовым синтаксисом объектов Javascript.

В настоящее время rosjs - это большая библиотека, поддерживающая множество сложных функций для визуализации и взаимодействия со сложными алгоритмами манипулирования и навигации на основе ROS. Однако его можно использовать и для управления роботом посредством предоставления пользователю web интерфейса управления (рис. 24).

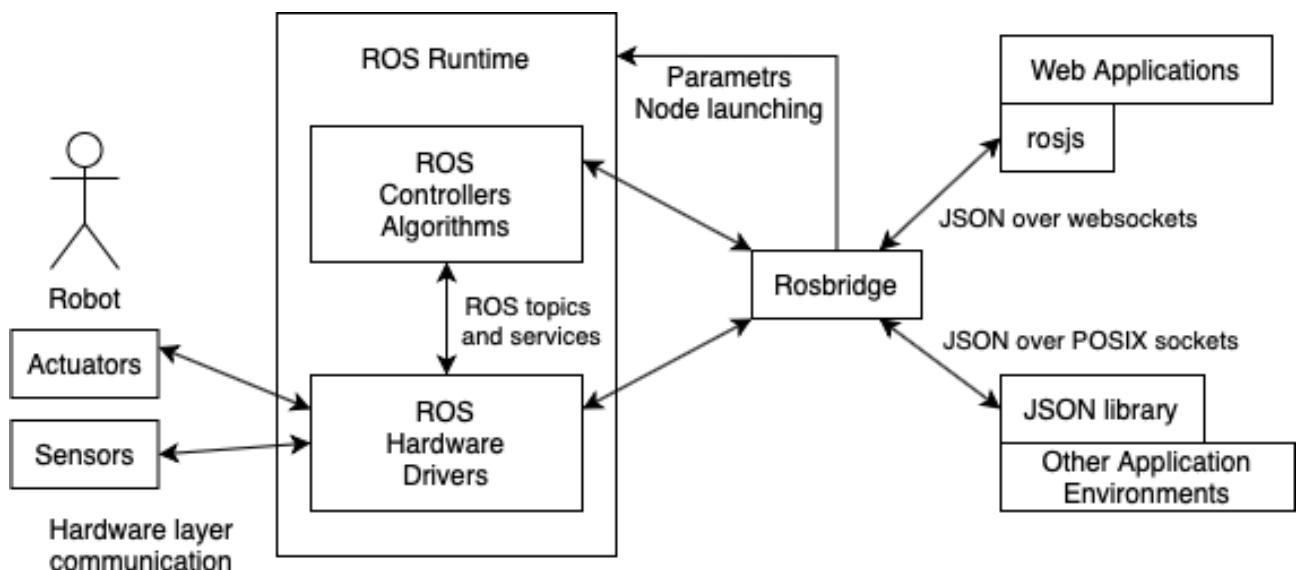


Рисунок 24 – Rosbridge: ROS для non-ROS приложений

### 3.6 GAZEBO

Симуляция робота является важным инструментом при разработке. Хорошо продуманный симулятор позволяет быстро тестировать алгоритмы, проектировать роботов, проводить регрессионное тестирование и обучать систему искусственного интеллекта с использованием реалистичных сценариев. Для симуляции походки робота-гексапода был выбран GAZEBO, который имеет возможность точно и эффективно моделировать роботов в сложных внутренних и наружных условиях.

Первым шагом в моделировании в ROS является создание трехмерной модели. 3D-модель может быть создана с помощью любого из инструментов моделирования, таких как Inventor, Fusion 360, 3dsMax. Когда модель готова,

она экспортируется в формат stl для интеграции в среду ROS.

Важно помнить, что все части робота, имеющие различную степень свободы передвижения, должны создаваться отдельно. Например, при создании корпуса с звеньями ног их надо экспорттировать отдельно, поскольку корпус статичен, а звенья ног динамичны и имеют определенную степень свободы для движения.

После создания трехмерных моделей отдельных компонентов они объединяются в ROS в окружения. Это возможно с помощью пакетной модели робота, которая содержит URDF-файлы синтаксического анализатора (Unified Robot Description Model). Также описания URDF связывают различные mesh (структурные сборки трехмерных моделей, состоящей из полигонов) через концепцию связей и соединений. Две части, соединенные joint (сустав / соединение), называются родительской и дочерней связью, а тип перемещения каждого элемента в модели определяется типом соединения между двумя связями. Тип joint может быть любым из следующих:

- Fixed – как следует из названия, этот тип является фиксированным и не имеет никаких степеней свободы;
- Revolute – при данном типе есть возможность вращения вокруг оси, движение ограничено в указанных верхних и нижних пределах;
- Continuous – схоже с Revolute и также имеет возможность вращения вокруг оси, за исключением отсутствия границ;
- Prismatic – скольжение вдоль оси в указанных пределах;
- Planar – движение перпендикулярно указанной оси;
- Floating – все шесть степеней свободы свободны.

Файлы URDF содержат описание моделей и роботов для импорта в ROS. Такое описание включает в себя название объекта, его mesh-структуру, оболочку для расчета столкновения, дополнительную информацию о визуальных компонентах объекта, таких как цвет, текстура и другие. Рассмотрим более подробную композицию urdf-файла:

- Тег <robot> содержит имя робота, которое будет отображаться во всех подсистемах ROS. Этот тег содержит описание всего окружения или его части (например, робота, объекта или датчика);
- Тег <link> также содержит имя и представляет визуальный компонент моделируемой среды или объект, который связан с другими

объектами с помощью соединений (теги `<joint>`). Внутри тега `<link>` можно увидеть описание визуальной части модели (тег `<visual>`), которая содержит информацию о геометрии объекта `<geometry>`. Возможные значения – `<cylinder>`, `<box>`, `<sphere>` и `<mesh>`;

- Тег `<origin>` устанавливает начальную позицию объекта в пространстве.
- Тег `<inertial>` определяет массу объекта и смещенный центр масс внутри объекта. Он необходим для симуляции гравитации;
- Тег `<collision>` определяет параметры столкновения. Это возможно благодаря тому, что ROS создает невидимый слой поверх визуального объекта, который является оболочкой, имеющей возможность сталкиваться с другими объектами;
- Тег `<joint>` - это соединение между двумя частями объекта `<link>`. Тег `<joint>` содержит описание родительского объекта и дочернего объекта (их имен), тип соединения, то есть тип движения (revolute – вращение вокруг некоторой оси с верхним и нижним пределом, continuous – бесконечный поворот вокруг определенной оси без ограничений и другие виды движения), также есть информация о местонахождении joint и информация о контроллерах.

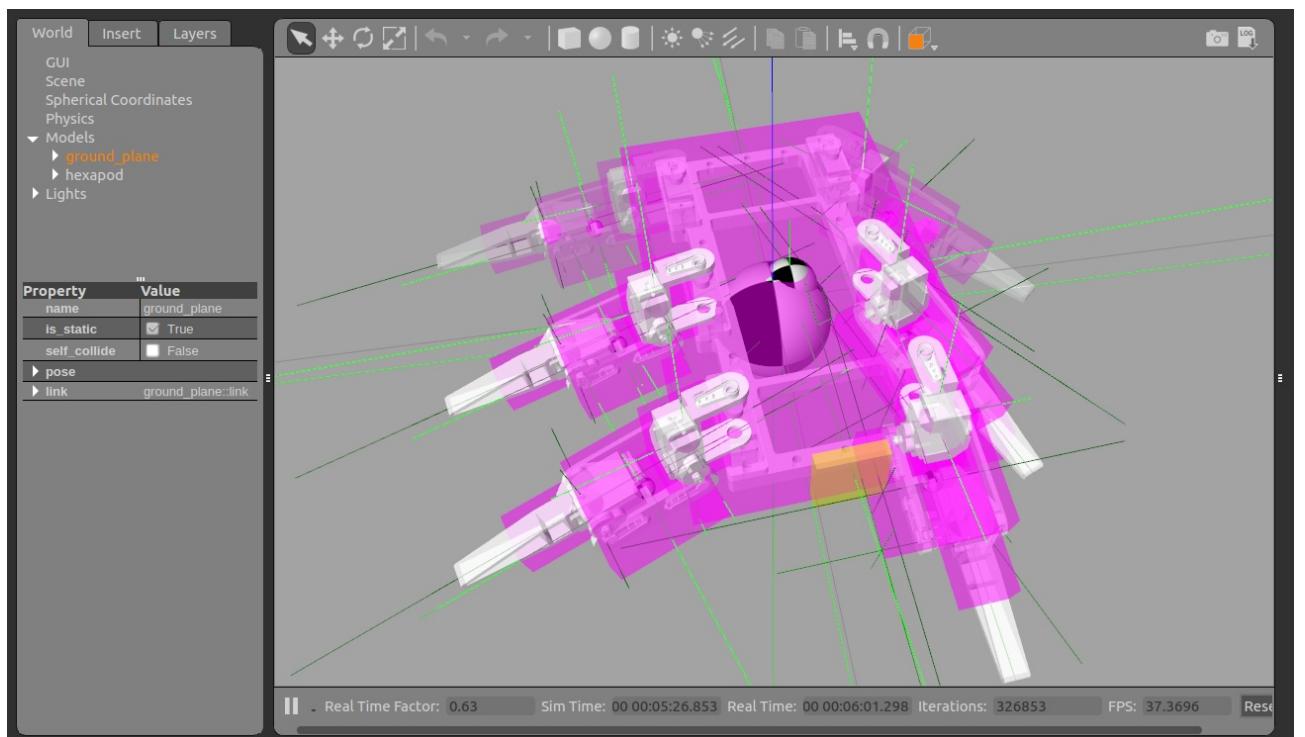


Рисунок 25 – Отображения моментов инерции и центра масс робота-гексапода в GAZEBO

Таким образом, файл URDF позволяет сделать полнофункциональную модель. Файл URDF обычно создается в текстовом редакторе. Конечно, в этом случае невозможно отслеживать в реальном времени изменения в трехмерном пространстве. Однако ROS имеет специальный пакет Rviz, который позволяет импортировать файлы URDF и показывает модель в трех измерениях. Кроме того, Rviz дает возможность манипулировать любыми объектами, входящими в среду. Например, перемещая ползунок, можно двигать ногами робота-гексапода. Rviz может анализировать столкновения и представлять их в виде трехмерной модели, отображать информацию о соединениях, массе и центре масс. Также некоторую информацию можно отображать и в GAZEBO (рис. 25). Для запуска пакета Rviz необходимо создать файл запуска, в котором описаны все узлы и их параметры.

### 3.7 Выводы по главе

Таким образом, была разработана система управления инсектоморфным роботом-гексаподом, на основе архитектуры, спроектированной во второй главе работы. Реализовано управление как посредством кросплатформенного web интерфейса, так и аппаратного джойстика.

Реализованы узлы ROS, которые позволяют осуществить взаимодействие компонентов управления и использовать ROS модули для визуализации, управления и настройки робота.

Предложенная реализация может быть расширена, путем добавления новых функциональных модулей, а также использована для управления новыми экспериментальными моделями. Подход в данном случае остается общим: ROS используется как связующий слой для интерфейсов оператора и системы генерации походки (в данном случае ФРУНД).

## 4 ТЕСТИРОВАНИЕ СИСТЕМЫ УПРАВЛЕНИЯ РОБОТОМ-ГЕКСАПОДОМ

В данной главе приводится тестирование решения полученного в ходе выпускной бакалаврской работы.

Для полной проверки работоспособности реализованной системы управления роботом-гексаподом необходимо:

- Протестировать адаптивность и кросплатформенность web интерфейса;
- Протестировать синхронизацию параметров между разными экземплярами интерфейса;
- Различные походки, путем управления как с web интерфейса, так и с аппаратного джойстика.

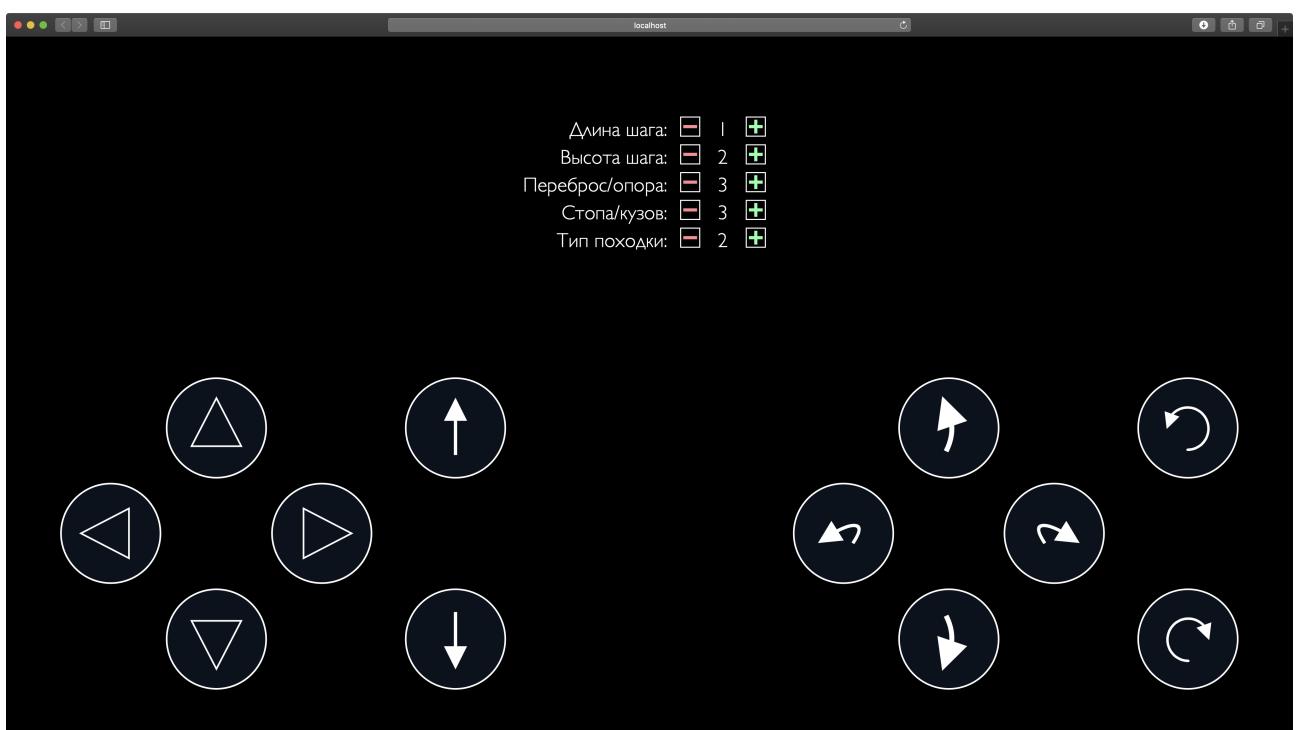


Рисунок 26 – Тестирование web интерфейса управления на компьютере

Тестирование адаптивности и кросплатформенности при разработке web интерфейса управления проводилось в браузере Safari на Apple iMac 27 с разрешением экрана 5120 x 2880, а также на Apple iPhone X с разрешением экрана 2436×1125. Как можно увидеть на рисунках 26 – 28, интерфейс успешно адаптируется к различному разрешению экрана, а также к ориен-

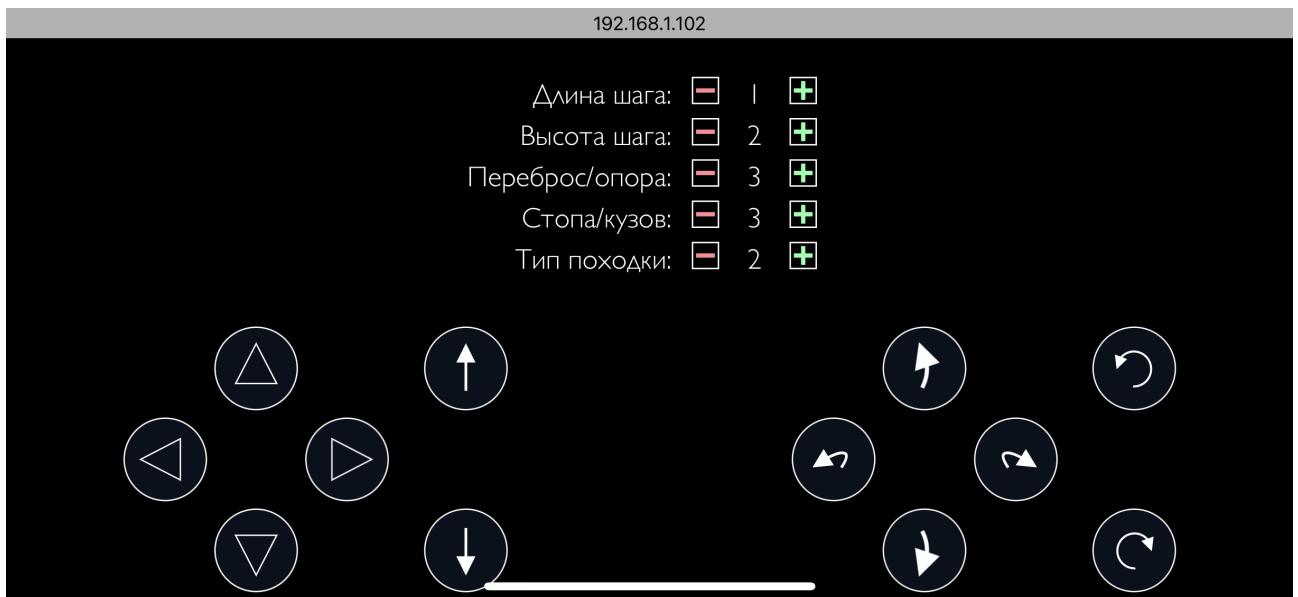


Рисунок 27 – Тестирование web интерфейса управления на смартфоне  
(горизонтальная ориентация)



Рисунок 28 – Тестирование web интерфейса управления на смартфоне  
(вертикальная ориентация)

тации экрана. Также из тестов видно, что синхронизация параметров между интерфейсами выполняется стablyно.

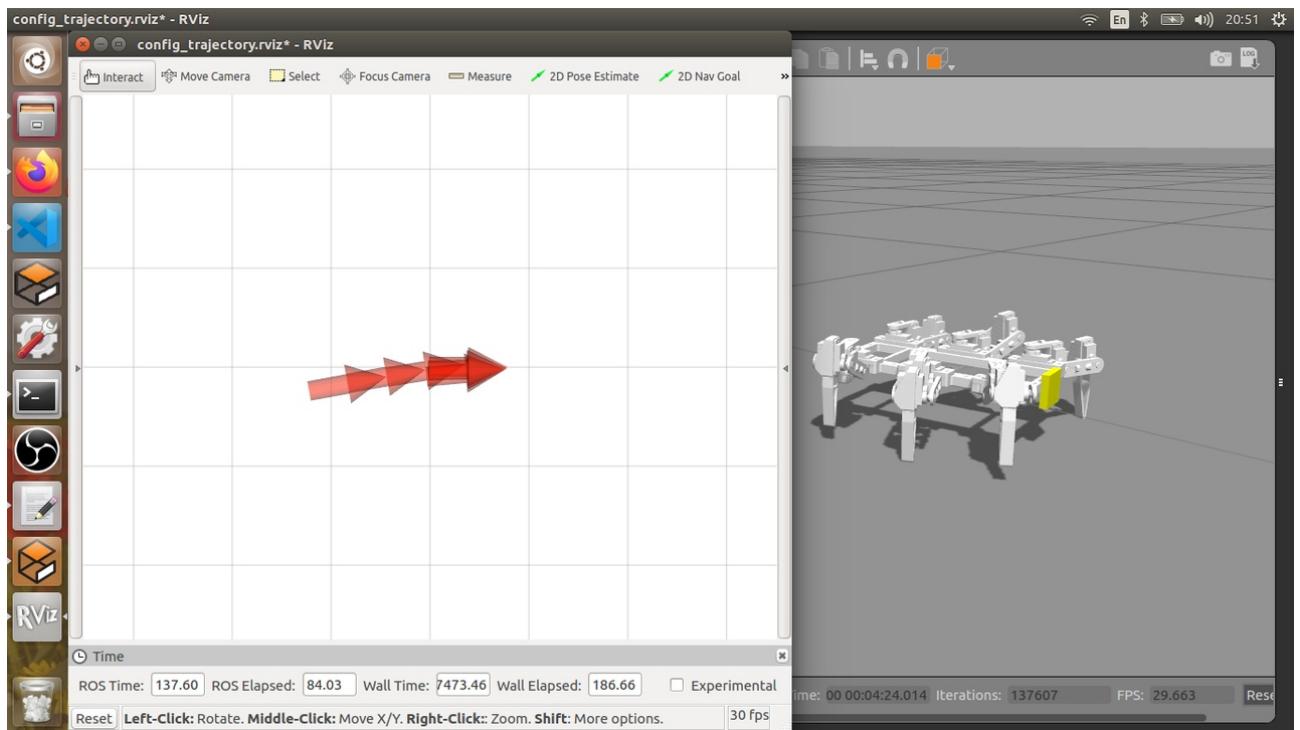


Рисунок 29 – Тест походки прямо

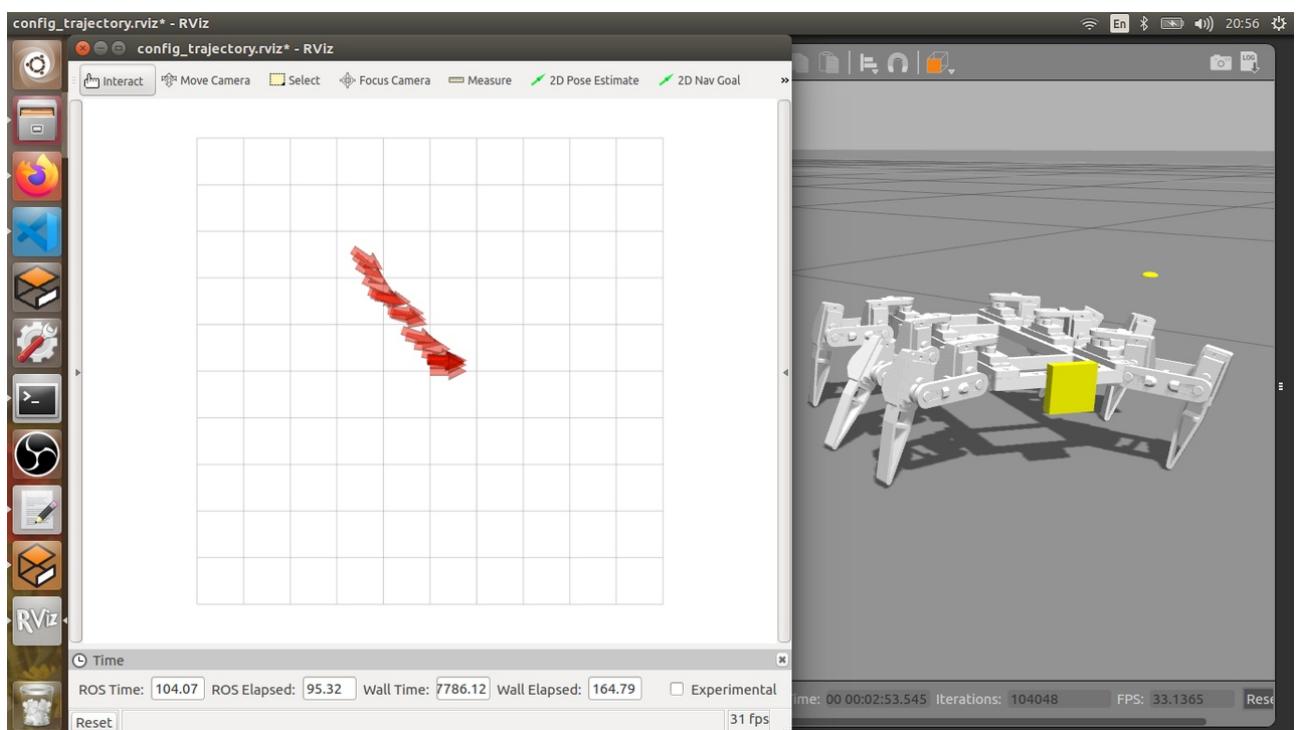


Рисунок 30 – Тест походки по диагонали

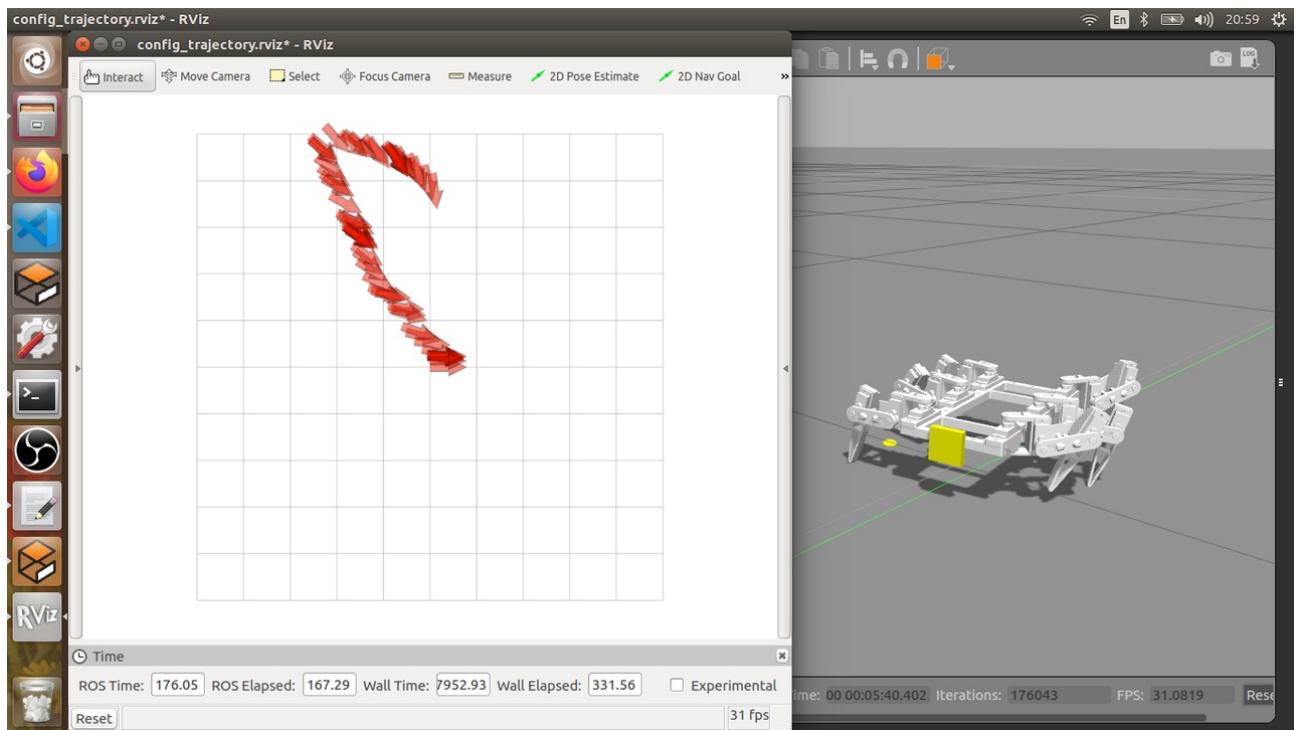


Рисунок 31 – Тест походки с разворотом

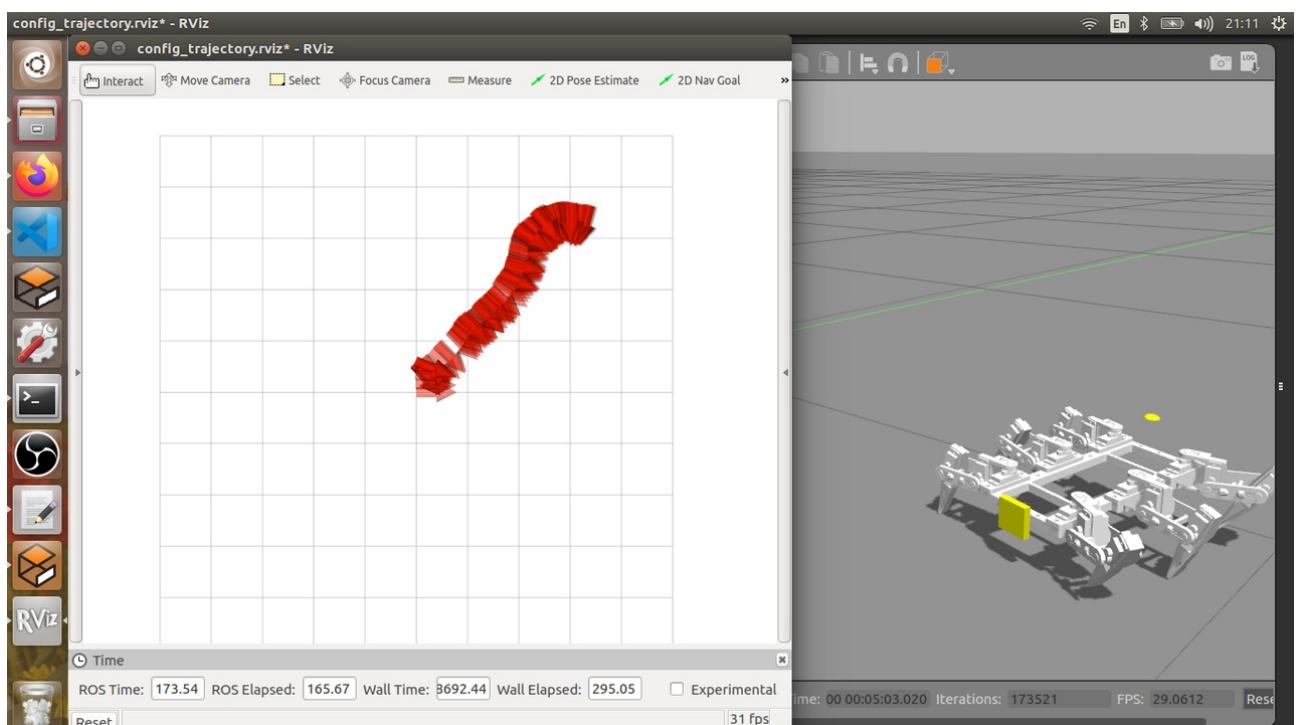


Рисунок 32 – Тест походки боком и поворота вокруг своей оси

## 4.1 Выводы по главе

В данной главе были проведены тесты разработанной системы управления роботом-гексаподом. Как можно видеть на рисунках 29 – 32 различные типы походок при управлении оператором выполняются успешно, есть небольшая погрешность, выражаяющаяся в незначительном отклонении от траектории, ввиду неидеальности модели и различных факторов, влияющих на походку. Управление как с web интерфейса, так и с аппаратного джойстика работает стабильно, без перебоев. Условия адаптивности, кросплатформенности, а также синхронизации между экземплярами интерфейсов управления, согласно поставленной задачи, также успешно прошли все необходимые испытания.

## ЗАКЛЮЧЕНИЕ

В данной работе была поставлена и достигнута цель, связанная с реализацией системы управления роботом-гексаподом.

В ходе работы были рассмотрены технические средства, необходимые для создания данной системы, основные парадигмы проектирования систем управления роботами. Проведен анализ и разбор аналогичного решения с выделением положительных и отрицательных моментов и путей усовершенствования разрабатываемой системы.

На основе изучения существующих систем управления, был выбран необходимый подход для управления инсектоморфным роботом, с использованием системы ФРУНД в качестве генератора траекторий движения.

Используя данный подход был реализован кросплатформенный адаптивный web интерфейс управления, модули осуществляющие связь параметров, получаемых от аппаратного и виртуального пультов, системы ФРУНД и модулей управления виртуальным и аппаратным роботом-гексаподом.

Были проведены эксперименты по тестированию решения, полученного в ходе выпускной бакалаврской работы, которое по результатом проявило полное соответствие поставленной задаче.

## СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Синтез локомоции шагания антропоморфного робота. / А.С. Горобцов [и др.] // Робототехника и искусственный интеллект: материалы VII Всероссийской научно-технической конференции с международным участием (г. Железногорск, 11 декабря 2015 г.) / под науч. ред. В.А. Углева. – Красноярск: Сиб. федер. ун-т, 2016. – С. 8-12.
2. Дистанционно управляемые подрывные машины семейства Sd.Kfz.302/303 Goliath (Германия) [Электронный ресурс] // Военное обозрение. – Режим доступа: <https://topwar.ru/94692-distancionno-upravlyayemye-podryvnye-mashiny-semeystva-sdkfz302-303-goliath-germaniya.html> (дата обращ. 16.03.2020).
3. Фокин, В. Г. Обзор и перспективы развития мобильных шагающих робототехнических систем / В. Г. Фокин, С. В. Шаныгин // Молодой ученый. – 2015. – №18. – С. 207-215.
4. Autonomous Locomotion and Navigation of Anthropomorphic Robot / A.K. Титов, А.Е. Марков, А.В. Скориков, П.С. Тарасов, А.Е. Андреев, С.С. Алексеев, А.С. Горобцов // Creativity in Intelligent Technologies and Data Science. Second Conference, CIT&DS 2017 (Volgograd, Russia, September 12-14, 2017): Proceedings / ed. by A. Kravets, M. Shcherbakov, M. Kultsova, Peter Groumpos ; Volgograd State Technical University [et al.]. – [Germany]: Springer International Publishing AG, 2017. – P. 242-255. – (Ser. Communications in Computer and Information Science; Vol. 754).
5. GAZEBO [Электронный ресурс]: офиц. сайт. – Режим доступа: <http://gazebosim.org> (дата обращ. 02.04.20).
6. ROS: an open-source robot operating system / M. Quigley, K. Conley, B. Gerkey et al // IEEE ICRA Workshop on Open Source Software – 2009. - Vol. 3.
7. Черноножкин, В. А. Система локальной навигации для наземных мобильных роботов / В. А. Черноножкин // Научно-технический вестник

- информационных технологий, механики и оптики. – 2008. – №57. – С. 13-22.
8. Инерциальная навигационная система [Электронный ресурс] // Системы навигации и позиционирования. – Режим доступа: <https://glonass-std.ru/sistemy-navigatsii/inertsialnaya-navigatsionnaya-sistema.html> (дата обращ. 12.04.2020).
  9. ФРУНД [Электронный ресурс]: офиц. сайт. – Режим доступа: <http://frund.vstu.ru> (дата обращ. 13.04.20).
  10. Ковальчук, А. К. Принципы построения программного обеспечения системы управления антропоморфным шагающим роботом. / А.К. Ковальчук, Д.Б. Кулаков, С.Е. Семенов // Автоматизация и современные технологии. – Москва, 2007. № 2.
  11. Vue.js the Progressive JavaScript Framework [Электронный ресурс]: офиц. сайт. – Режим доступа: <https://vuejs.org> (дата обращ. 15.04.20).
  12. ROSBridge\_suite – ROS Wiki [Электронный ресурс]: офиц. сайт. – Режим доступа: [http://wiki.ros.org/rosbridge\\_suite](http://wiki.ros.org/rosbridge_suite) (дата обращ. 16.04.20).
  13. Brown, J. H. How fast is fast enough. Choosing between Xenomai and Linux for real-time applications / H. Brown, B. Martin. [Электронный ресурс] – 2010 – Режим доступа: <https://www.osadl.org/fileadmin/dam/rtlws/12/Brown.pdf> (дата обращ. 22.05.20).
  14. Edwards, S. ROS-industrial: applying the robot operating system (ROS) to industrial applications / S. Edwards, C. Lewis // IEEE Int. Conference on Robotics and Automation. - ECHORD Workshop. - St. Paul, 2012.
  15. Development of the insectoid walking robot with inertial navigation system / Б.А. Егунов, А.Л. Качалов, М.К. Петросян, П.С. Тарасов, Е.В. Янкина // Proceedings of the 2018 International Conference on Artificial Life and Robotics (ICAROB2018) (February 1-4, 2018, B-CON Plaza, Beppu, Oita, Japan) / Editor-in- Chief Masanori Sugisaka ; International Steering Committee of International Conference on Artificial Life and Robotics (ICAROB), ICAROB society (ALife Robotics Corporations Ltd.), IEEE Fukuoka Section (Japan). – [Japan], 2018. – P. 54 (Mobile Robotics ; OS7-2).

16. The Walk-Man Robot Software Architecture [Электронный ресурс] // ResearchGate. – Режим доступа: [https://www.researchgate.net/publication/302916141\\_The\\_Walk-Man\\_Robot\\_Software\\_Architecture](https://www.researchgate.net/publication/302916141_The_Walk-Man_Robot_Software_Architecture) (дата обращ. 23.05.20).
17. Hayes-Roth, B. A blackboard architecture for control / B. Hayes-Roth // Artif. Intell. – 1985 (26). – P. 251–321. DOI: 10.1016/0004-3702(85)90063-3.
18. The Ach library: a new framework for real-time communication / N. Dantam, D. Lofaro, A. Hereid et al. // IEEE Robot. Autom. – 2014, Mag. 22, P. 76–85. DOI: 10.1109/MRA.2014.2356937.
19. Orebäck, A. Evaluation of architectures for mobile robotics / A. Orebäck, H. I. Christensen // Robots – 2003. – (14). – P. 33–49. DOI: 10.1023/A:1020975419546.
20. Magyar, G. Comparison study of robotic middleware for robotic applications / G. Magyar, P. Sinčák, Z. Krizsan // Emergent Trends in Robotics and Intelligent Systems (Cham: Springer International Publishing AG). – 2015. – Vol. 316. – P. 121–128.