Федеральное государственное автономное образовательное учреждение высшего образования

«Балтийский федеральный университет имени Иммануила Канта»

Университетский колледж

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

Разработка последовательности диагностики и ремонта неисправностей навигации сервисного робота

11.02.16 Монтаж , техническое обслуживание и ремонт электронных приборов и устройств

Разработал студент

группы РП51

\_\_\_\_\_\_\_\_\_\_\_А.Д. Доронин

Руководитель

\_\_\_\_\_\_\_\_\_\_А.В. Петрущенков

Консультанты:

\_\_\_\_\_\_\_\_\_\_И.О. Фамилия

\_\_\_\_\_\_\_\_\_\_И.О. Фамилия

Калининград

2023

**Содержание**

[Введение 4](#__RefHeading___Toc1583_2929712768)

[РАЗДЕЛ 1 ТЕХНИЧЕСКИЙ АНАЛИЗ НАВИГАЦИИ СЕРВИСНОГО РОБОТА 7](#__RefHeading___Toc3206_2556014353)

[1.1 Общие понятия 7](#__RefHeading___Toc3058_2556014353_Copy_1)

[1.2 Описание структурной схемы навигации сервисного робота 12](#__RefHeading___Toc3058_2556014353)

[1.3 Анализ неисправностей навигации сервисного робота 13](#__RefHeading___Toc1572_3523201741)

[1.4 Диагностика схемы, выявление неисправного узла по алгоритму 14](#__RefHeading___Toc1585_2929712768)

[1.5 Описание электрической принципиальной схемы навигации сервисного робота 15](#__RefHeading___Toc1587_2929712768)

[РАЗДЕЛ 2 КОНСТРУКТОРСКО - ТЕХНОЛОГИЧЕСКИЙ РАЗДЕЛ 19](#__RefHeading___Toc1803_3158998568)

[2.1 Определение условий эксплуатации навигации сервисного робота 19](#__RefHeading___Toc3066_2556014353)

[2.2 Описание конструкции навигации сервисного робота 19](#__RefHeading___Toc3068_2556014353)

[2.3 Разработка последовательности технологического процесса ремонта навигации сервисного робота 35](#__RefHeading___Toc3082_2556014353)

[2.4 Выбор и обоснование выбора основных и вспомогательных материалов для ремонта навигации сервисного робота 35](#__RefHeading___Toc3084_2556014353)

[2.5 Выбор и обоснование оборудования и оснастки для ремонта навигации сервисного робота 37](#__RefHeading___Toc3086_2556014353)

[2.6 Организация охраны труда при выполнении последовательности ремонта навигации сервисного робота 37](#__RefHeading___Toc3088_2556014353)

[3. ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ РАЗРАБОТАННОЙ ПОСЛЕДОВАТЕЛЬНОСТИ РЕМОНТА НАВИГАЦИИ СЕРВИСНОГО РОБОТА 40](#__RefHeading___Toc3090_2556014353)

[3.1 Расчет нормы времени и трудоемкости на диагностику и ремонт навигации сервисного робота 40](#__RefHeading___Toc3092_2556014353)

[3.2 Расчет фонда заработной платы и отчислений 40](#__RefHeading___Toc3094_2556014353)

[3.3 Расчет стоимости основных материалов и комплектующих изделий 41](#__RefHeading___Toc3096_2556014353)

[3.4 Расчет накладных расходов и прочих расходов, влияющих на стоимость ремонта навигации сервисного робота 41](#__RefHeading___Toc3098_2556014353)

[3.5 Расчет цены услуги ремонта навигации сервисного робота 41](#__RefHeading___Toc3100_2556014353)

[Заключение 42](#__RefHeading___Toc1589_2929712768)

[Список литературы 43](#__RefHeading___Toc1591_2929712768)

[Приложение А 45](#__RefHeading___Toc1576_3523201741)

[Приложение Б 46](#__RefHeading___Toc3507_2929712768)

[Приложение В 47](#__RefHeading___Toc3509_2929712768)

# Введение

Сервисные роботы - это автоматизированные устройства, которые помогают людям, выполняя рутинные, удаленные, опасные или повторяющиеся задачи, такие как уборка помещений. Как правило, они автономны или управляются интегрированной системой управления и могут управляться вручную.

Одной из ключевых систем сервисного робота является его система навигации, которая находится на стыке между механизмами, датчиками и программной составляющей.

Данная тема была выбрана по причине увеличения влияния робототехники на жизнь практически каждого современного человека. Когда ранее действительно встречаемая в жизни робототехника ограничивалась ЧПУ-станками и простыми киосками, а болшинство сложных роботов были больше исследовательскими и образовательными, то сейчас же сервисные роботы распространены гораздо больше.

В качестве примера можно привести роботов курьеров, которых уже можно встретить на улицах, где они самостоятельно доставляют грузы, пересекая черзвычайно сложный городской ландшафт, где могут встречаться совершенно непредсказуемые препятствия.

**Объектом исследования являются сервисные роботы различных направлений, а в частности различные виды навигации, локализации построения маршрута, а также методы диагностики и устранения неисправностей в вышеупомянутых системах.**

**Предметом исследования будет типовой навигационный стэк на основе ROS, а также обучающий робот Turtlebro производства компании Voltbro, реализующий его, его устройство модуля навигации, а также выявление методов диагностики**

****Целью** работы является выработка общих мероприятий по диагностике и ремонту навигационной системы сервисного робота с помощью реального сервисного робота, а также обобщение и оформление полученной информации**

**Задачи ВКР:**

- выявить типовые несправности навигационной системы

- изучить различные алгоритмы локализации и навигации, основные методы получения и передачи данных

- разработать методическое руководство по ремонту

- проанализировать небходимый набор инструментов, расходников, программного обеспечения, а также их финансовую составляющую

- систематизировать полученные данные

Для достижения поставленных задач будут применены следующие **методы исследования.**

- анализ литературы, в частности технической документации. Также будет затронута теоретическая составляющая алгоритмов навигации

- анализа нормативно-правовой или нормативно-справочной документации, рабочих чертежей по теме проекта (работы);

- изучение и обобщение отечественной и зарубежной практики;

исходные коды проектов, реализующих различные подсистемы навигации сервисных роботов

- теоретический анализ,

- конкретизация,

- индукция и дедукция,

- классификация и обобщение полученной информации в легко воспринимаемый методичным формат

**Научная новизна** данной работы заключается в сближении программной и аппаратной части. Также большое количество трудов выпускается по теме разработки сервисных роботов и их узлов, но гораздо меньше касательно их ремонта и подготовки квалифицированного персонала. Предполагается использование глубокого понимания программного устройства навигационной системы для нахождения оптимальных и продуктивных методов диагностики и ремонта. Также у данной работы есть высокий образовательный потенциал в связи с применением обучающего робота, удобного в разборе и изучении.

**Практическая значимость** в потенциальной пользе и прибыльности поставленного на поток ремонта сервисной робототехники, а также применения накопленных знаний, навыков и материальной базы для укрепления экономики через увеличение эффективности труда и уменьшения зависимости от зарубежных технологий через импортозамещение.

**Структура ВКР** предполагается таковой:

1. Обозрительное исследование основных методов, алгоритмов, задействованных узлов и реализаций навигации сервисного робота
2. Выявление обощенных потенциальных мест появления неисправности
3. Выработка последовательности устранения каждой из перечисленных неисправностей
4. Составление списка необходимых расходников, инструментария, а также немало важно: програмнного обеспечения необходимого для корректной диагностики и ремонта.
5. Экономическое обоснование выбранных методов и материалов
6. Заключение, а также приложения с полезными материалами

В пределах данной работы основным фокусом будет фреймворк разработки роботов ROS (Robot Operation System). Выбор обоснован наличием в колледже роботов на этой платформе, а основным принципам данного фреймворка. Он идеально подходит для как изучения робототехники, так и создания собственных систем (в частности робот моего с товарищами авторства BigBang).

Концепты и опыт этого фремворка можно легко перенести на другие системы, с помощью которых могут быть созданы различные

# **РАЗДЕЛ 1 ТЕХНИЧЕСКИЙ АНАЛИЗ НАВИГАЦИИ СЕРВИСНОГО РОБОТА**

## **1.1 Общие понятия**

Для начала стоит выделить основные типы сервисных роботов, так как каждое конкретное применение предъялвяет собственные требования к навигации

Международная организация по стандартизации определила "сервисных роботов" как роботов, выполняющих задачи, полезные для людей и оборудования, за исключением приложений промышленной автоматизации. Согласно стандарту ISO 8373, роботы должны обладать "степенью автономности".

Это "способность выполнять намеченные задачи, основываясь на своем текущем состоянии и ощущениях, без вмешательства человека". В сервисных роботах эта степень варьируется от частичной автономии, которая включает в себя взаимодействие человека и робота, до полной автономии, которая представляет собой отсутствие активного вмешательства человека.

Хотя существует широкий спектр применения роботов, которые помогают человеку выполнять задачи, в настоящее время эти роботы можно разделить на несколько категорий (см. Таблица 1).

С точки зрения программирования - для того, чтобы рассматривать конкретные ситуации и реализации необходимо выбрать платформу (фреймворк) разработки. Мною был выбран фреймворк Robot Operating System (ROS).

Таблица 1. Классификация сервисных роботов

|  |  |
| --- | --- |
| **Сервисные** | Сервисные роботы - это системно автономные, адаптивные интерфейсы, которые взаимодействуют, общаются и предоставляют услуги клиентам организации. |
| **Бытовые** | Бытовые роботы выполняют задачи, которые люди регулярно выполняют в непроизводственной среде, например, подметают полы, стригут газоны и обслуживают бассейны. |
| **Научные** | Роботизированные системы выполняют множество функций, начиная от повторяющихся задач с использованием генных пробоотборников и секвенаторов и заканчивая системами, которые заменяют ученых при разработке и проведении экспериментов, анализе данных и формулировании гипотез. От морских глубин до космоса, эти научные роботы выполняют задачи, которые человеку трудно или невозможно выполнить автономно. |
| **Промо-роботы** | Сфера услуг начинает использовать роботов для взаимодействия с клиентами и посетителями мероприятий. Роботы могут стать отличным источником взаимодействия. Примерами использования роботов на мероприятиях для привлечения посетителей являются робот-фотограф «Eve» или «Промобот». |

### Robot Operating System (ROS)

Robot Operating System (далее ROS) - это гибкая платформа (фреймворк) для разработки программного обеспечения роботов. Это набор разнообразных инструментов, библиотек и определенных правил, целью которых является упрощение задач разработки ПО роботов.

Создание действительно надежного, универсального программного обеспечения для роботов чрезвычайно сложная задача. С точки зрения робота, проблемы, которые кажутся тривиальными для людей, часто требуют очень сложных технических решений. Часто разработка такого решения не под силу одному человеку.

ROS была создана, чтобы стимулировать совместную разработку программного обеспечения робототехники. Каждая отдельная команда может работать над одной конкретной задачей, но использование единой платформы, позволяет всему сообществу получить и использовать результат работы этой команды для своих проектов.

Мета-операционная система не похожа на обычную операционная систему, такие как Windows, Linux и Android. ROS работает "поверх" существующей операционной системы. Для работы ROS необходима базовая операционная система Linux (например дистрибутив Ubuntu).

После завершения установки ROS на OC Linux можно использовать функции, предоставляемые обычной операционной системой. В дополнение к основным функциям, предоставляемым Linux, ROS обеспечивает дополнительный функционал, необходимый для роботов. Например: работа с библиотеками, передача / прием данных для разных устройств, планирование и обработка ошибок. Этот тип программного обеспечения также называется промежуточным программным обеспечением (middleware ) или программным фреймворком.

Таблица 2. Наиболее активные платформы для создания робототехники

| MSRDS10 | Microsoft Robotics Developer Studio, Microsoft - U.S. |
| --- | --- |
| ERSP11 | Evolution Robotics Software Platform, Evolution Robotics - Europe |
| **ROS** | **Robot Operating System, Open Robotics12 - U.S.** |
| OpenRTM | National Institute of Adv. Industrial Science and Technology (AIST) - Japan |
| OROCOS | Europe |
| OPRoS | ETRI, KIST, KITECH, Kangwon National University - South Korea |

### Почему ROS?

ROS обеспечивает разработчиков библиотеками и инструментами для создания приложений робототехники. ROS обеспечивает аппаратную абстракцию, предлагает драйверы устройств, библиотеки, визуализаторы, обмен сообщениями, менеджеры пакетов и многое другое. ROS выпускается в соответствии с условиями BSD лицензии (подавляющее большинство пакетов) и с открытым исходным кодом.

На платформе ROS также создан робот Turtlebro, который имеется в колледже в наличии, с использованием которого проводились несколько конкурсов WorldSkills. Использование его в качестве пимера сделает работу более понятной и ближе к практике.

### Что дает готовая платформа

1. **Повторное использование программных модулей**. Разработанный программный модуль, легко запускается и переиспользуется в любом другом приложении. Вопросы установки зависимостей и других библиотек хорошо проработан и автоматизирован.
2. **Готовый протокол коммуникации** Основная проблема комплексных робототехнических систем, это решение задач коммуникации в рамках одного приложения. Для решения этих задач ROS содержит все необходимые утилиты. Любой программных модуль может быть представлен как отдельный процесс, взаимодействующий с другими процессами по сетевому протоколу. Такой подход позволяет создавать независимые и простые в повторном использовании программные модули, которые возможно запустить/остановить/модифицировать на любом устройстве.
3. **Развитость средств разработки и отладки** ROS предоставляет готовые инструменты для отладки, инструмент 2D-визуализации (rqt), и инструмент 3D-визуализации (RViz), инстумент 3D симуляции (Gazebo).
4. **Активное и открытое сообщество** Сообщества разработчиков робототехники из академического мира и промышленности, были относительно закрытыми до последнего времени. Но сейчас мы видим активное, и главное открытое сотрудничество всех участников. В центре этого изменения -- программная платформа с открытым исходным кодом. В случае ROS существует более 5000 пакетов, которые были разработаны и выложены в общий доступ. Описание этих пакетов, инструкций и другой полезной информации -- превышает 18 000 Wiki страниц.
5. **Собственная экосистема** Вокруг ROS сформирована собственная экосистема (по аналогии с платформами Android и Apple). В ней существуют разработчики аппаратных платформ, разработчики программных модулей, энтузиасты и компании производители промышленного оборудования, единое место распространения и хранения готовых модулей, тысячи станиц документации. Все участники взаимодействуют и работают в рамках единой платформы.

## **1.2 Описание структурной схемы навигации сервисного робота**

Система навигации (иначе называемая «навигационный стек») - важнейшая система, необходимая для их корректной работы. Исполнения разнятся в сложности и функциональности в зависимости от изделия.

Стек представлен в схематичном виде в [**Приложении 1**](#_toc914). Навигационный стек начинается с аппаратной части. В случае робота BigBang взаимодействием с аппаратной частью занимается микроконтроллер Arduino Mega 2560 (/arduino\_interface).

Далее будут рассмотрены основные элементы навигационного стека и их типовые реализации , что позволит эффективно подбирать конкретные методики и инструмент диагностики неисправностей. Подробный разбор будет выполнен в [**разделе 2.2**](#_toc708).

1. Аппаратный уровень (электроника)
2. Аппаратный уровень (механика)
3. Уровень прошивки
4. Уровень одометрии
5. Коммуникационный уровень
6. Сервер(а) карты
7. Уровень локализации
8. Уровень планировщика(ов) маршрута

Стоит также понимать, что данный стек – лишь самая основа робота, с помощью которой производится абстракция управления механизмами и движения для релаизации конечных задач каждого устройства.

Помимо перечисленных этапов в роботе также может быть реализована различная высокоуровневая логика, которая решает конкретные бизнес-задачи. С помощью алгоритмов, ии, нейронных сетей и т. д. Данный уровень не входит в пределы данной работы.

## **1.3 Анализ неисправностей навигации сервисного робота**

При поиске неисправности необходимо двигаться «по стеку» снизу вверх. Т. е. Независимо от неисправности проверку всех систем необходимо выполнять начиная с аппаратного уровня и заканчивая самым высоким программным.

Для создания алгоритма типичного порядка диагностики неисправности будет рассмотрена реальная проблема, возникшая на роботе turtlebro во время соревнования WorldSkills в компетенции обслуживание сервисных роботов.

### Возможные проблемы в порядке стека

1. Проблемы с питанием: Робот может не заряжаться, не держать заряд, не включаться или выключаться сам по себе. Это может быть вызвано неисправным аккумулятором, поврежденным портом зарядки или проблемами с зарядным устройством
2. Проблемы с механикой: Робот может иметь повреждения механических частей, таких как руки, кисти, шарниры или другие механизмы Это может быть вызвано неправильнымMoebius 4 Channels Motor Driver Shield for Arduino MEGA2560 использованием робота, падениями или другими физическими повреждениями
3. Проблемы с движением: Робот может не двигаться, двигаться с трудом, двигаться в неправильном направлении или двигаться слишком быстро или медленно. Также робот может некорректно обрабатывать получаемую обратную связь с приводов движения (неисправные энкодеры, обрыв проводов датчиков и т. д.
4. Проблемы с прошивкой: Как и в любом программном обеспечении в прошивке робота (программе на микроконтроллере) могут присустствовать ошибки. Обычно они проявляются редко, но важно понимать, что нужно выходить на контакт с поставщиком для устранения подобных неисправностей.
5. Проблемы с сенсорами: Робот может не реагировать на окружающую среду, давать ложные сигналы или неправильно интерпретировать данные от сенсоров Это может быть вызвано повреждением сенсоров, проблемами с программным обеспечением или электроникой
6. Проблемы с коммуникацией: Робот может не соединяться с другими устройствами, не обмениваться данными или давать неправильные ответы Это может быть вызвано проблемами с Wi-Fi, Bluetooth или другими протоколами связи, неправильной настройкой или электроникой
7. Проблемы с программным обеспечением: Робот может не выполнять команды, давать неправильные ответы, зависать или выключаться сам по себе Это может быть вызвано ошибками в программном обеспечении, неправильной настройкой или обновлением программного обеспечения
8. Проблемы с безопасностью: Робот может представлять угрозу для безопасности окружающих, например, если он не остановится при обнаружении препятствий, не будет выполнять команды безопасности или будет выполнять неправильные команды

## 1.4 Диагностика схемы, выявление неисправного узла по алгоритму

При поиске неисправности важно понимать хотя бы на базвоом уровне архитектуру конкретного робота, а также платформы, на которой он сделан. Во многих современных фреймворках предусмотрены инструменты отладки как для кода, так и для системы в целом.

Основной инструмент – средства визуализации процессов, а также немаловажно – логгирования. При начале поиска неисправности надо выяснить в первую очередь: какой модуль системы вышел из строя?

Но для начала стоит обратить внимание, что делать на основе одного пункта заключительный вывод ни в коем случае нельзя. В такой сложной системе, как сервисный робот важно проверить все основные варианты, затем как можно больше потенциальных вариантов причины неисправности и только потом уже проверять гепотезу. Дело в том, что легко перепутать возможную и реальную причину неисправности, а в поспешных попытках ее диагностирования и исправления создать условия, лишь усложняющие поиск настоящей неисправности.

Далее будет предложен типовой порядок диагностики неисправности. В случае робота на основе ROS можно воспользоваться средством визуализации rqt, а именно инструментами topic monitor, tf tree, node graph для визуализации процессов. Для роботов реализованных на других платформах можно воспользоваться средствами той платформы. Главное, что стоит понять: этот алгоритм легко расширяем и применим ко многим системам (для людей с опытом в этих системах)

Проблема, решаемая в данном вопросе – робота заносит на одно колесо:

1. Приближенно определеить неисправный узел, начинать с уровня, на котором впервые проявилась проблема.   
   В данном примере: При попытке установки точки движения на карте робот достигает цели, но постоянно корректируется изза того, что его “заносит”
2. Попытаться локализиировать проблему – воссоздать минимальные условия ее воспроизведения.   
   В данном примере: Далее предпринята попытыка напрямую поуправлять роботом, посылая команды в соответствующий топик (/cmd\_vel), робот едет, но все также быстрее одним из двух колес
3. Далее повторятся пункт 2, но с внимательным мониторингом логов и визуализацией данных.   
   В данном примере: По логам все в порядке, а вот данные одометрии ведут себя странно. А именно, по ним не видно, чтобы робота заносило, напротив – будто бы они оба дают одинаковые показания, и лишь за счет лидара и верхнеуровневой локализации он восстанавливает свои настоящие координаты
4. Проверить все этапы стека навигации, начиная с самого нижнего и заканчивая этапом минимального воспроизвдеения  
   В данном примере: Проверено состояние аккумуляторов, проверены все контакты двигателей, а их провода подключения на предмет обрывов иил плохих контактов. Все исправно.
5. Сделать предположение и проверить его. Затем повторять пока не будет найдена причина.  
   В данном примере: Ситуация похожа на неисправный энкодер мотора, точно проверить их можно посмотрев на равномерность, последовательность и количество тиков энкодера при вращении колеса используя двухканальный осциллограв. Обнаружено, что энкодеры одного из моторов срабатывают через раз.
6. Решить найденную проблему  
   В данном примере: Замена платы энкодера или всего мотора в зависимости от модели.

## 1.5 Описание электрической принципиальной схемы навигации сервисного робота

В [Приложении Б](#_toc969) представлена принципиальная электрическая схема робота BigBang собственной разработки. При его проектировании и изготовлении были изучены несколько моделей сервисных роботов, а именно: TurtleBro производства Voltbro, Reset производства SkolTech и Setup от Лаборатории робототехники Сбербанка.

Все три робота выполнены на базе Robot Operating System. TurtleBro позиционируется именно как сервисный робот, с возможностью гибкого расширения для различных целей, например экскурсовод и т.д. Роботы Reset и Setup предназначены для выступления в соревновании Eurobot Open, которое характеризуется ограниченным временем и трудностью заданий.

Наш робот BigBang создан по примеру всех троих, заимствуя идеи, но предлагая свои решения. Далее будет представлено описание электрического устройства робота, а в частности элементов участвующих в локализации и навигации.

### Питание и проводка

В качестве источника питания используется литий-полимерный аккумулятор с номинальным напряжением 3C (11.1 В), емкостью 5000 mAh, токоотдачей 25С (250 А) и пиковой краткосрочной токоотдачей 50С (500 А). Для подзарядки используется блок питания с балансировкой. На входе установлен предохранитель на 10 А.

Сам аккумулятор находится в огнеупорном мешке. Робот имеют: кнопка общего питания, кнопка эктренного отключения всей силовой электроники.

Для питания двигателей и помпы использутеся понижающий DC-DC преобразователь с уставкой на 12В. Для питания сервоприводов используется такой же преобразователь, но с выходным напряжением 5.3 вольта.

Питание для двигателей разведено через плату-расширитель для Arduino Mega производства Moebius. На данной плате расширения присутствуют контакты для подключения до 4 шлейфов для двигателя (используется 3). Подробнее о данной плате будет написано в следующем пунтке.

Помпа и воздушный клапан используют питание 12 вольт, которое коммутируется через два релейных модуля, запитанных от понижающего преобразователя цепи сервоприводов и управляемых с микроконтроллера.

Питание Raspberry Pi 4 происходит через еще один отдельный преобразователь с двумя USB 5В выходами на 3А суммарной нагрузки через кабель USB-Type C. Также от него питается ЖК-дисплей бортового компьютера через кабель USB-A to microUSB.

Питание лидара RPLidar-S1 происходит через USB-порт Raspberry Pi 4, проходя через USB-TTL преобразователь, который преобразует USB-порт в последовательный RS-232. Скорость соединения: 256000 бод.

Для связи Raspberry Pi и Arduinoиспользуется кабель USB-A to USB-B. В качестве интерфейся подключения экрана используется кабель HDMI.

### Микроконтроллер

В качестве низкоуровневого микроконтроллера используется Arduino Mega 2560. Правктически в каждом роботе на ROS происходит разделение обязанностей между основным комьютером и МК (микроконтроллером). Это сделано для абстракции управления различными элементами робота. Зачастую управление элементами выполнено в виде конечного автомата. В случае робота BigBang МК отвечает за:

1. Управление коллекторными двигателями постоянного тока через драйвер (H-мост)
2. Счет “тиков” энкодеров на вале мотора для подсчета пройденных расстояний за один “шаг” навигации
3. Управление аналоговыми сервоприводами через I2C плату расширения PCA9685. Всего 16 каналов с 12 битным ШИМ-генератором (4096 разрешение ШИМ сигнала).
4. Интерфейс для I2C LCD-дисплея для вывода подсчитанного результат матча.
5. Контакты для подключения аксселерометра, который пока что не применяется, но в ближайшем будуещм вероятно будет интегрирован для увеличения точности локализации, следовательно и навигации
6. Остальная дополнительная электроника, например контакт запуска робота (через чеку, замыкающую подтянутый к 5В цифровой вход на землю) и т.д. по необходимости

МК управляет всеми низкоуровневыми элементами робота, делая это через удобны для программиста интерфейс. У помпы также имеется свой LC-фильтр, который позволяетс избежать помех накоммуникационных линиях при включении

Как упоминалось в прошлом разделе для удобства сборки используется плата-расиширитель Moebius (Приложение В). В данной плате разведены удобные группы контактов для подключения различных устройств по интерфейсу I2C, аналоговых сервоприводов и других.

Основной причиной выбора данной платы стал встроенный H-мост для управления тремя коллекторными двигателями. В шлейфе из шести контактов присутствуют следующие сигналы и питания:

1. Первый полюс мотора
2. +5В VCC энкодеров
3. Сигнал первого энкодера
4. Сигнал второго энкодера
5. GND
6. Второй полюс мотора

### В контексте ремонта

Подробное описание электрической схемы робота BigBang полезно тем, что содержит в себе все основные модули, которые присутствуют на типовом сервисном роботе, что позволяет делать предположения и выводы, основываясь на нее.

Также немаловажно разнообразие интерфейсов, применяемых в роботе, и инжнерных решений связанных с электроникой. Разнообразные интерфейсы предрасположены к различным проблемам, таким как интерференции.

Например решение добавить LC-фильтр на помпу появилось после первых испытательных пусков, когда при включении помпу, особенно на нагрузку в виде груза микроконтроллер перезагружался изза сильной помехи по питанию. С того времени электрическая схема была сильно доработана.

# РАЗДЕЛ 2 КОНСТРУКТОРСКО - ТЕХНОЛОГИЧЕСКИЙ РАЗДЕЛ

## **2.1 Определение условий эксплуатации навигации сервисного робота**

Специалисту-технику, занимающимся ремонтов и эксплуатацией сервисных роботов необходимы знания механики и электроники - для ремонта модульных роботов различных технических материалов и документов, связанных с и понимание процесса приемки роботов и робототехники и того, как технического обслуживания роботов и робототехники на протяжении всего производственного процесса. операций.

Другие навыки включают в себя инновационное мышление и способность применять аналитические навыки во всех сферах деятельности. и способность разрабатывать решения сложных задач по техническому обслуживанию и эксплуатации.

Техник по робототехнике должен разбираться в новейших технологиях, используемых для создания и управления сервисными роботами и уметь применять эти знания на практике.

### Организация работы и безопасность

<куски описания специальности>

Системное администрование

Следование инструкции

Эксплутация в пределах спецификации

## 2.2 Описание конструкции навигации сервисного робота

### Платформы (фреймворки) в робототехнике

В последнее время, в области робототехники особое внимание уделяется платформам. Понятие платформа обычно резделяют на программную платформу и аппаратную платформу. Программная платформа для роботов включает в себя набор инструментов, которые используются для разработки ПО роботов.

Можно выделить типовые, задачи программной платформы: работа с низкоуровневыми устройствами, аппаратная абстракция и коммуникация, навигация, распознавание образов, управление и установка пакетов и зависимостей, подключение библиотек, инструменты для отладки и разработки.

Аппаратные платформы, включают в себя готовые исследовательские и образовательные устройства (TurtleBot, TurtleBro). А также готовые промышленные системы.

Важно отметить, что аппаратные платформы совместимы с программными платформами, что позволяет разрабатывать прикладные программы не имея опыта работы с оборудованием и не тратя время на его разработку.

Совместимость интерфейсов и методов взаимодействия с оборудование, позволило огромному количеству разработчиков ПО внести свой вклад в развитие робототехники.

Унифицированные интерфейсы и методы работы с устройствами позволяют накапливать и обмениваться готовыми решениями всему сообществу заинтересованных людей в робототехнике.

Навигация сервисного робота может быть поделена на прорграммную и аппаратная составляющую

Аппаратное обеспечение (датчики, элементы управления) :

1. Энкодеры
2. Лидар
3. Лазерные / ультразвуковые / инфракрасные дальнометры
4. Веб Камера - конкретно здесь не задействована
5. Драйвера
6. Микроконтроллеры
7. Бортовой компьютер (возможны различные сочетания)

Программное обеспечение:

1. Пограммные модули навигации
   1. Построение маршрута
   2. Локализация
   3. Обработчик одометрии
   4. Набор программных параметров (причина: унификация программных модулей со специлазией путем установки параметров, также цель - кастомизация)
   5. Логгирование
   6. Архивирование

### Локализация по методу Монте-Карло

Алгоритм локализации по методу Монте-Карло (MCL) используется для оценки положения и ориентации робота. Алгоритм использует известную карту окружающей среды, данные датчика дальности и данные датчика одометрии. Чтобы узнать, как создать объект и использовать этот алгоритм, см. раздел monteCarloLocalization.

Чтобы локализовать робота, алгоритм MCL использует фильтр частиц для оценки его положения. Частицы представляют собой распределение вероятных состояний для робота. Каждая частица представляет собой возможное состояние робота. Частицы сходятся вокруг одного места, когда робот перемещается в окружающей среде и воспринимает различные части окружающей среды с помощью датчика дальности. Движение робота регистрируется с помощью датчика одометрии.

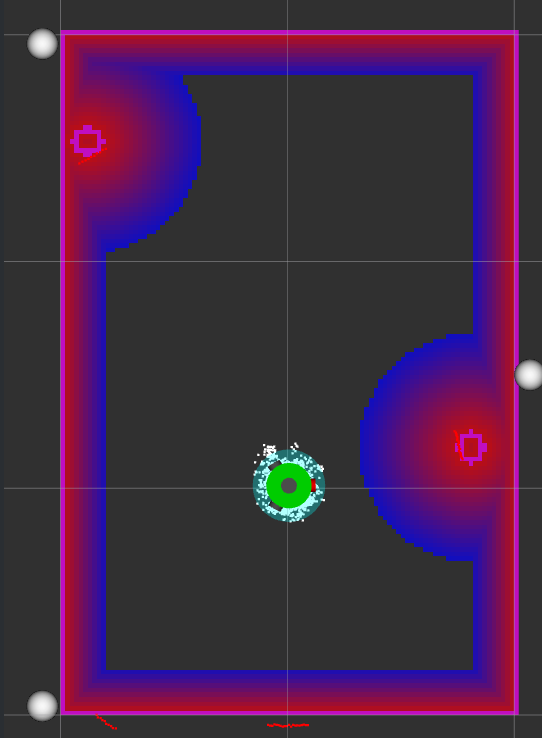
В ходе этого процесса частицы обновляются:

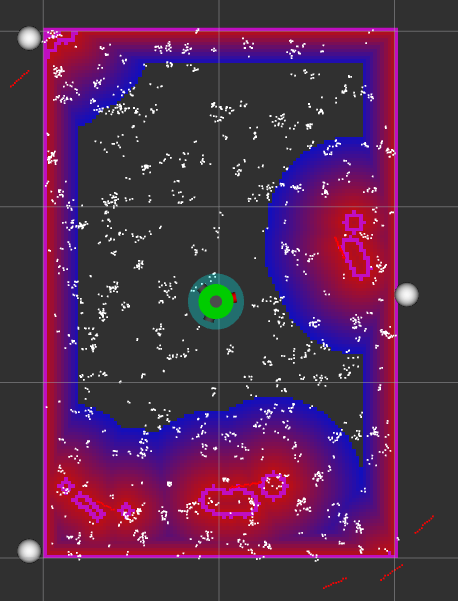
Частицы распространяются на основе изменения позы и заданной модели движения модели движения. Чстицам затем присваиваются веса, основанные на вероятности получения показаний датчика дальности для каждой частицы. Это значение основано на модели датчика, указанной модели сенсора.

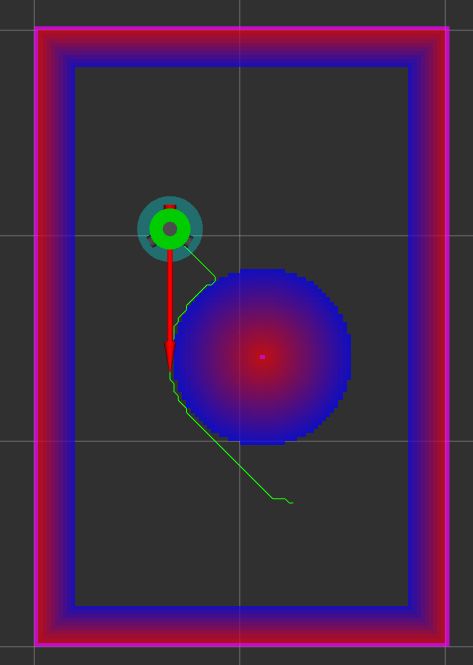
На основе этих весов извлекается оценка состояния робота на основе весов частиц. Группа частиц с наибольшим весом используется для оценки положения робота.

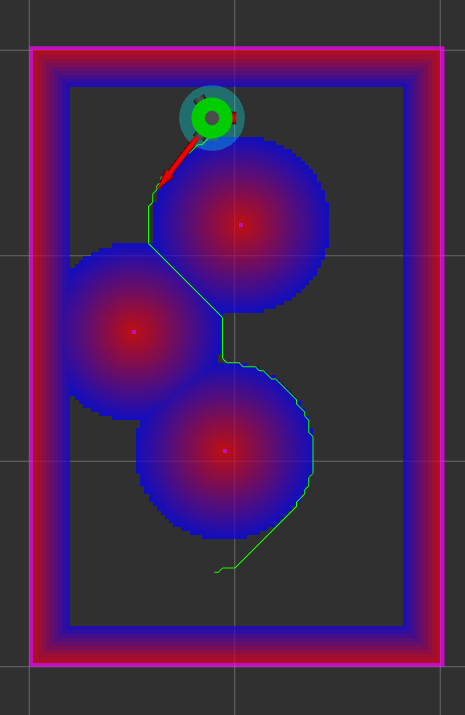
Наконец, частицы подвергаются повторной выборке на основе указанного интервала. Повторная выборка регулирует положение частиц и повышает производительность за счет регулировки количества используемых частиц. Это ключевая функция для адаптации к изменениям и сохранения актуальности частиц для оценки состояния робота.

Алгоритм выводит оценочную позицию и ковариацию. Эти оценки представляют собой среднее значение и ковариацию кластера частиц с наибольшим весом.

Рисунок 1. Репрезентация состояния робота через возможные (точки)

Рисунок 2. Локализация при потере опорных точек

Рисунок 4.Объезд простого препятствия, обратите внимание на оптимизацию пути (красная стрелка) через комбинацию двух алгоритмов

Рисунок 3.Построение более сложного маршрута

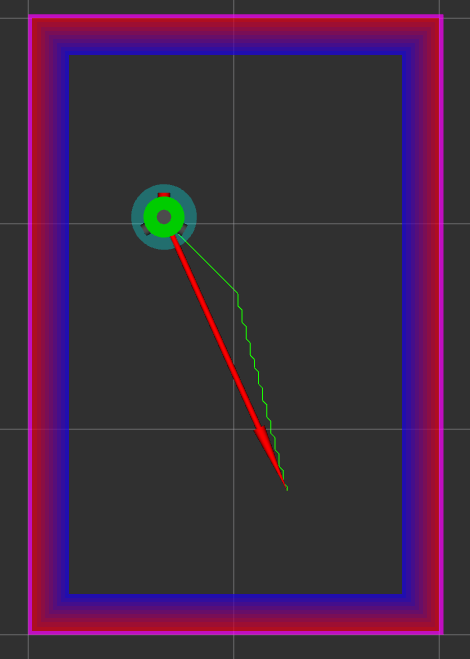
В большинстве систем построения исполбзуется так называемая “карта стоимости” (costmap). В области мобильных роботов, основанных на платформах ROS (Robot Operating System)

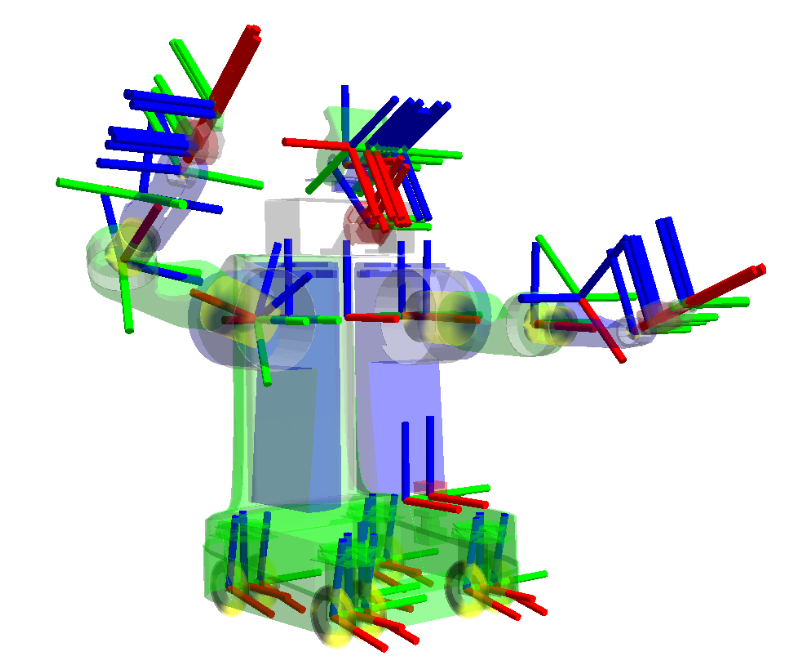
, для получения кратчайшего пути широко используются A\*, Dijkstra и BFS (поиск в ширину). Однако такой

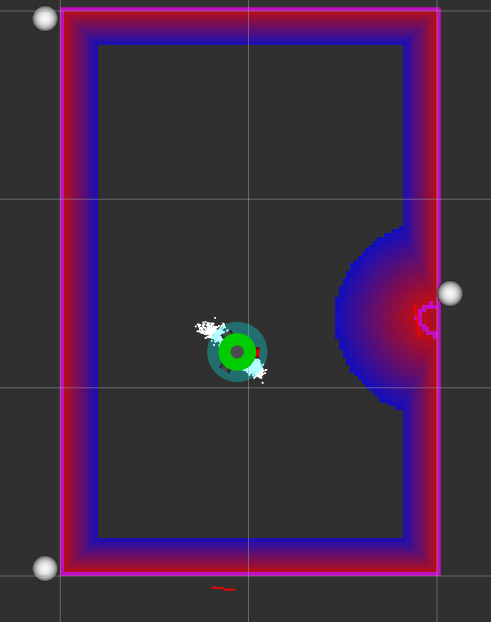
подход часто приводит к тому, что пути проходят близко к препятствиям, расположенным в углах. В

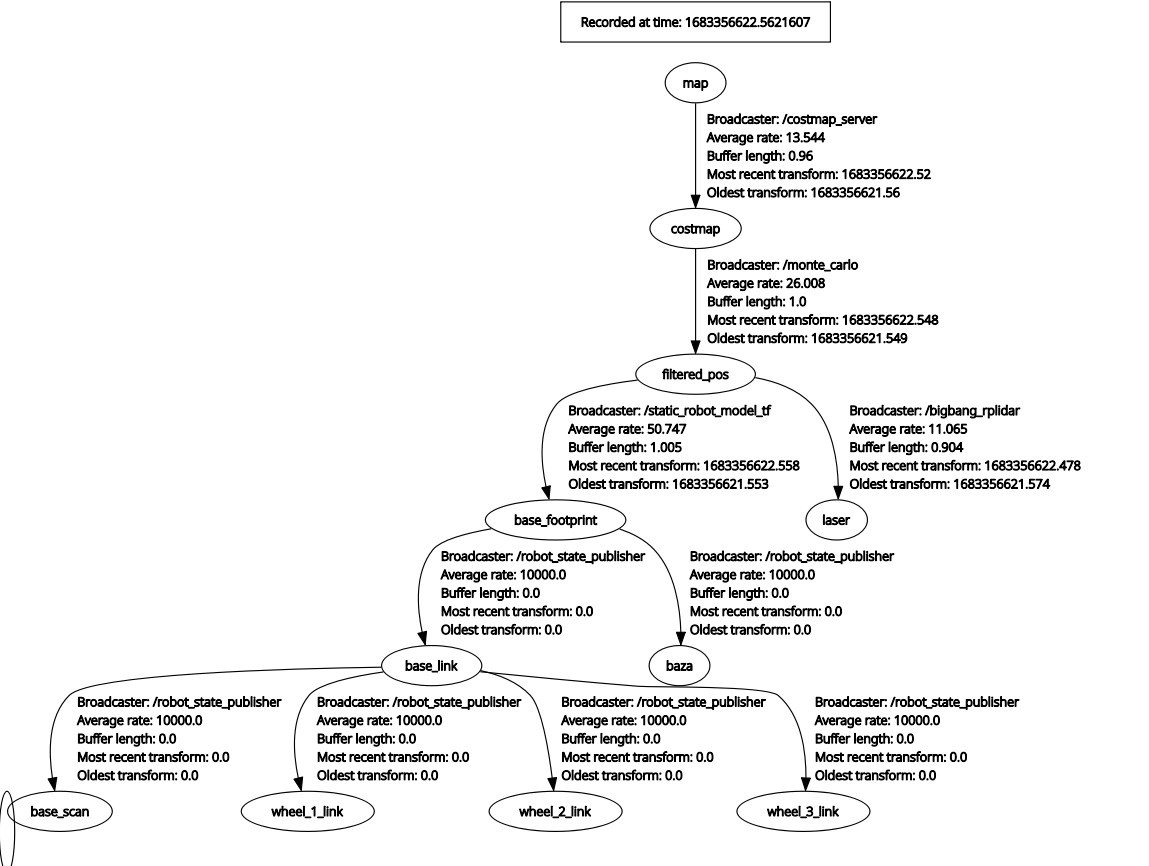
практических приложениях кумулятивные ошибки позиционирования в процессе навигации возникают из-за

старения компонентов, заноса колес и так далее. Кроме того, локальное планирование траектории робота полностью соответствует глобальному планированию траектории, поэтому робот может легко пересечь инфляцию карты затрат и сталкиваться с препятствиями при повороте. Решением может являться изменение траектории поворота робота путем добавления виртуальных препятствий к препятствиям в углах карты. Глобальный путь генерируется алгоритмом планирования пути вдоль виртуального препятствия, чтобы робот мог двигаться по гладкой дорожке, удаленной от препятствия при повороте. Данный алгоритм гарантирует, что робот остается вдали от физических препятствий и избегает столкновений.

Рисунок 5. Построение простого маршрута

Рисунок 6. Визуализация фреймов робота. Без иерархической структуры вычисления перемещений отедльных элементов были бы в разы сложнее

Рисунок 7. Распределение точек при одной опорной точке

Рисунок 8. Навигационные фреймы робота BigBang

В комплексных системах навигации зачастую, хотя и в немного разных реализациях присутствует система навигационных “фреймов” (иначе называемых “рамками”). В случае ROS реализацией данной системы занимается библиотека tf (кратко от transfrom).

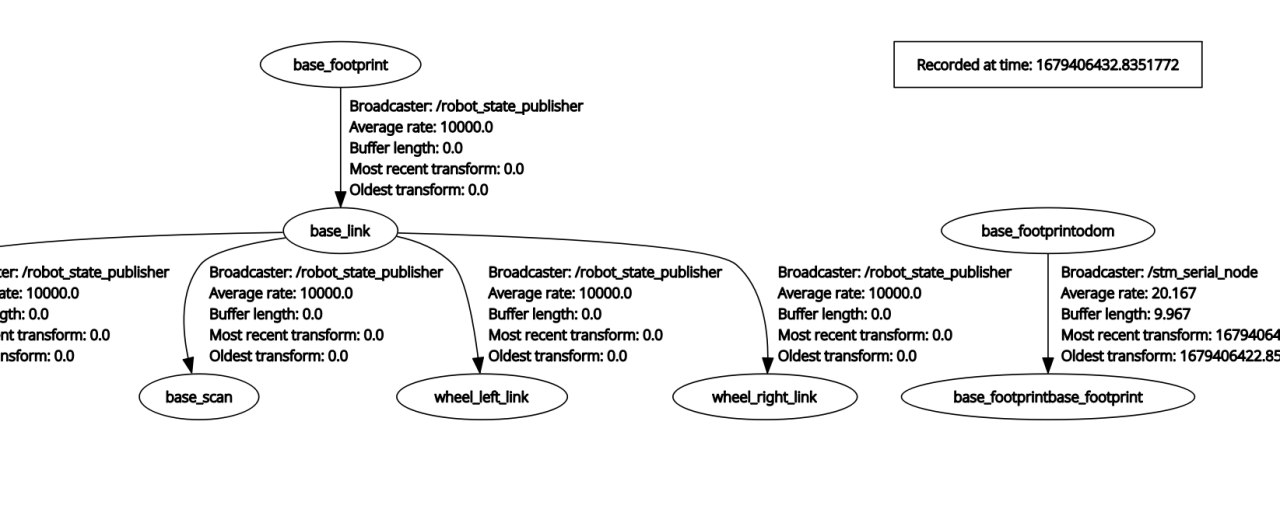
Библиотека tf была разработана для обеспечения стандартного отслеживания рамок координат и преобразования данных рамках всей системы таким образом, чтобы пользователи отдельных компонентов быть уверены, что данные находятся в нужной им системе координат,требуя знания всех рамок координат системе.

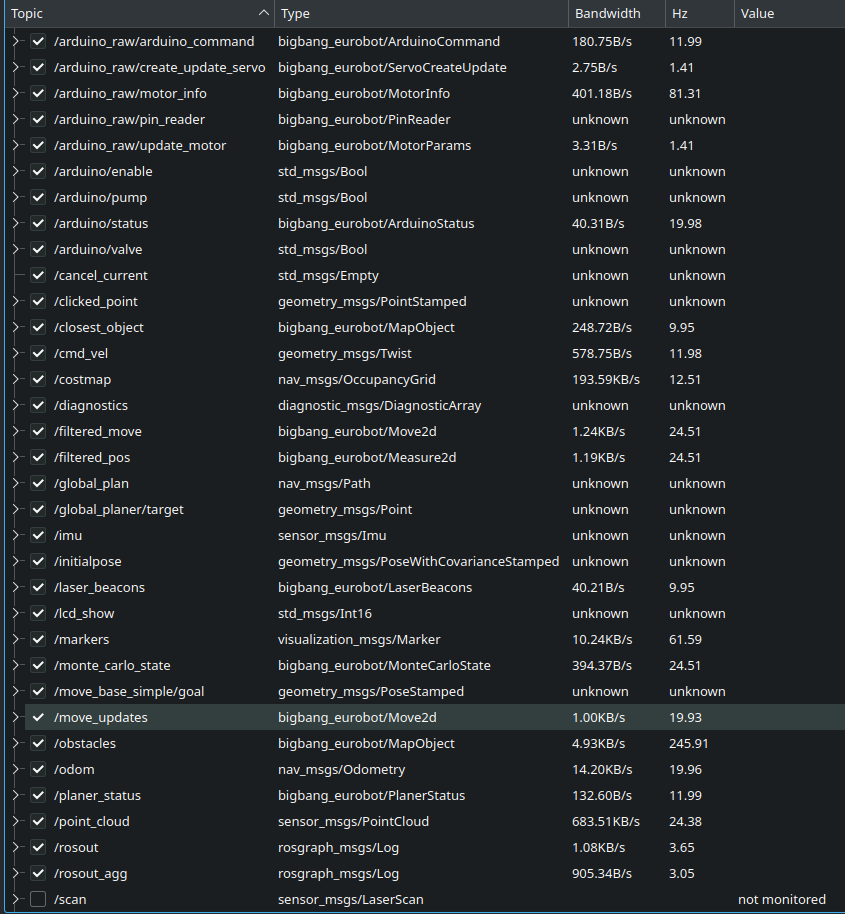
Во время ранней разработки робота система (ROS), отслеживающая фреймы координат была определена как общая проблема для разработчиков.

Сложность этой задачи сделала ее обычным местом для ошибок, когда разработчики неправильно применяли преобразования к данным. То также является сложной из-за часто распределенных источников

информации о преобразованиях между различными наборами систем

координат.

Рисунок 9. Пример "разбитого" дерева фреймов смещений (tf tree). Обратите внимание на причину: некорректное имя двух фреймов

Рисунок 10. Доступные топики, частота обновления и сетевая нагрузка

Стоит учесть тот факт, что обмен сообщениями происходит по TCP/IP каналам, соответственно при удаленном подлкючении или соединении нескольких устройств в сеть, что может быть необходимо при создании сложной сети, нужно учитывать состояние сети.

Плохая локальная сеть – источник большинства проблем при обслуживании роботов. Поэтому очень важно уметь правильно и быстро подключать роботов к сети, а также иметь базовое представление о системном адмнистрировании, так как от этого напрямую зависит эффективность диагностики.

Коварность сетевых неисправность также заключается в том, что сетевые проблемы могу создавать ситуацию, которая скрывает настоящие проблемы.

Пример подобных ситуаций:

1) После некорректной настройки окружения (не экспортирована переменная окружения ROS\_IP) программа визуализации на первый взгляд работает корректно. Можно увидеть топики робота, фреймы и даже некоторые визуализировать. Но после запуска стека навигации и попытке дать команду на движение к точке на карте, ничего не происходит. Если не перепроверить корректность базовых настроек, то можно посчитать, что либо некорректно работает алгоритм, неправильные параметры навигации или проблема с аппаратурой (обрыв кабеля моторов и т.д.). В такой комплексной системе любой узел может быть причиной, поэтому важна внимательность и качественная методичка, чтобы избежать подобных проблем в будущем.

2) При удаленном подключении шаги настройки окружения сети были выполнены не на хост-устройстве, а в удаленном терминале самого робота (экспортирование некорректного ROS\_IP). Тогда при попытке что либо запустить из этого терминала будут выдаваться сообщение о невозмоности подлкючения к самому себе. Эта проблема гораздо более заметная, но все же неочевидная для человека мало знакомого с сетевой архитектурой робота.

## **2.3 Разработка последовательности технологического процесса ремонта навигации сервисного робота**

1) Проверка питания - батарея, регуляторы, физические дефекты

2) Проверка всех соединений - разрывы, заломы,

## **2.4 Выбор и обоснование выбора основных и вспомогательных материалов для ремонта навигации сервисного робота**

<Расходники для пайки>

**<Расходнии для работы с коннекторами и проводкой>**

**Специалист должен знать и понимать следующее**

- устройство, назначение и принципы работы контрольно-измерительного оборудования оборудование, используемое в их работе, и правила пользования этими приборами;

- Виды и классификацию ручных и измерительных инструментов; и

- Как пользоваться слесарными инструментами;

- Знать и выбирать болты, гайки, шпильки и крепежные изделия для выполнения конкретных задач; и

- Принципы использования профилактических измерительных приборов, таких как пирометры, инфракрасные камеры, виброанализаторы и т.д.

Техники должны уметь

- Использовать измерительные инструменты

- Выбирать соответствующий режущий инструмент для выполнения требуемой задачи

- Определять, выбирать и использовать соответствующие измерительные и контрольные инструменты - Использовать соответствующие режущие инструменты для выполнения требуемых задач.

Выбирать и использовать соответствующие измерительные инструменты

- Выбирать стопорные инструменты и приспособления для простой сборки и разборки

сборочных единиц и механизмов.

**- уметь выполнять шабрение, распиливание, пригонку и припасовку, притирку,**

доводку, полирование.

1. Запасные аппаратные модули навигации
2. Коннекторы интерфейсов шин данных

## **2.5 Выбор и обоснование оборудования и оснастки для ремонта навигации сервисного робота**

1) Мультиметр

2) Осциллограф

3) Прибор тестирования аккумуляторов

4) Персональный компьютер - программное обеспечение

5) Утилиты поставляемые производителем

6) Возможно логический анализатор? Для глубокой диагностики и ремонта конктреных модулей по типу лидара

## **2.6 Организация охраны труда при выполнении последовательности ремонта навигации сервисного робота**

<Классические правила работы с электроникой>

<Правила работы с ПК? Правила инф без?>

Сохранение бэкапов параметров, программных модулей

- Основные обозначения описаний создания программного обеспечения;

- Основные подходы к разработке и тестированию программного обеспечения

- Основные подходы к созданию алгоритмов поведения роботов

- Основные подходы к настройке и программированию конечных автоматов;

- Основы теории автоматического управления

- Основы языка программирования bash;

- Основы языка программирования Python/Cpp

- Базовые алгоритмы для дистанционного управления

- Базовые алгоритмы для автономных навигационных систем

- Базовые алгоритмы для систем компьютерного зрения

- Типичная архитектура программного обеспечения сложного робота

- Базовые знания по администрированию ОС Linux

- Основы программирования сценариев Linux

- Основы управления операционными системами роботов

- Основы управления операционными системами роботов; основы установки, конфигурирования и настройки операционных и метаоперационных систем. Пакеты операционной системы: основы установки, конфигурирования и настройки;

- Принципы создания и конфигурирования пакетов приложений

- Принципы работы системы контроля версий (VCS). Специалисты должны уметь:

- следовать инструкциям производителей аппаратного и программного обеспечения устанавливать, управлять и настраивать

- работать с документацией и искать в интернете решения возникших проблем работать с документацией и искать в интернете решения возникших проблем

- применять стандартные алгоритмы для установки, настройки и управления Программное и аппаратное обеспечение.

- изменять собственные алгоритмы для установки, конфигурирования и управления. управлять программным и аппаратным обеспечением для реализации поставленных задач. Задачи, поставленные перед специалистами.

- Решать сложные проблемы установки, конфигурирования и управления. Самостоятельно и независимо следовать инструкциям производителя программного и аппаратного обеспечения Самостоятельно и независимо следовать инструкциям производителя программного и аппаратного обеспечения;

- Устанавливать, настраивать и тестировать пакеты прикладных программ ROS. Linux;

- Распространенные системы контроля версий (GitHub, BitBucket, SourceForge и т.д.);

- Устанавливать и настраивать контроллеры и роботов. Общие системы контроля версий (например, GitHub, Bitbucket, SourceForge); для установки и настройки контроллеров и роботов. подключить систему;

- оптимизировать поведение компонентов и модулей робототехнической системы в соответствии с технической документацией. Оптимизировать поведение компонентов и модулей робототехнической системы в соответствии с технической документацией.

- Настроить алгоритмы управления робототехническим комплексом (автономная навигация). (например, автономная навигация, компьютерное зрение).

- Разрабатывать программы для микропроцессорных устройств и интерфейсных устройств. Интерфейсные устройства, используемые для модернизации робототехнического комплекса

- Разрабатывать программное обеспечение для робототехнических комплексов в соответствии с В соответствии с техническим заданием.

# **3. ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ РАЗРАБОТАННОЙ ПОСЛЕДОВАТЕЛЬНОСТИ РЕМОНТА НАВИГАЦИИ СЕРВИСНОГО РОБОТА**

## **3.1 Расчет нормы времени и трудоемкости на диагностику и ремонт навигации сервисного робота**

1) Диагностика аппаратной части

2) Диагностика программной части

## 3.2 Расчет фонда заработной платы и отчислений

1) Матералы

2) Инструменты

3) ПО

4) Зарплата

5) Курсы повышения квалификации

6) R&D

## 3.3 Расчет стоимости основных материалов и комплектующих изделий

1)

## **3.4 Расчет накладных расходов и прочих расходов, влияющих на стоимость ремонта навигации сервисного робота**

1)

## **3.5 Расчет цены услуги ремонта навигации сервисного робота**

1)

**Заключение** (выводы и рекомендации относительно возможностей применения полученных результатов)

**СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ**

# Заключение

# Список литературы

https://arxiv.org/abs/2105.12121

https://manual.turtlebro.ru/links

https://www.bachelorstudies.com/bachelor/robotics

https://etu.ru/assets/files/ru/5-100/kudrovo/docs/obrazovatelnaya-programma-dopolnitelnogo-obrazovaniya-tehnicheskoj-napravlennosti-robototehnika.pdf

https://sch354c.mskobr.ru/files/dop\_robototehnika\_bogacheva\_t\_p3.pdf

https://gaoyichao.com/Xiaotu//ros/refs/tf\_The\_Transform\_Library.pdf

https://publications.hse.ru/pubs/share/folder/gyswt4tx19/51170774.pdf

Нариньяни А. С., Телерман В. В., Ушаков Д. М., Швецов И. Е.  
Программирование в ограничениях и недоопределенные модели //  
"Информационные технологии", No7, 1998. – М., Издательство  
“Машиностроение”. – с. 13-22.  
2. Невдяев Л. «CDMA: сигналы и их свойства», 2000.  
http://www.osp.ru/nets/2000/11/141475/  
3. Ипатов В. П. «Широкополосные системы и кодовое разделение  
сигналов. Принципы и приложения». Москва: Техносфера, 2007. –  
488 с.  
4. Карпов В. Э. «О некоторых особенностях применения  
недоопределенных моделей в робототехнике». // Сборник научных  
трудов. Т.1. – М.: Физматлит, 2009. – с. 520-532.  
5. Платонова М. В. «Использование шумоподобных сигналов ИК-  
диапазона для системы навигации мобильных роботов» // Сборник  
«Мобильные роботы и мехатронные системы». – М.: Издательство  
Московского университета, 2009. – с. 148-155.

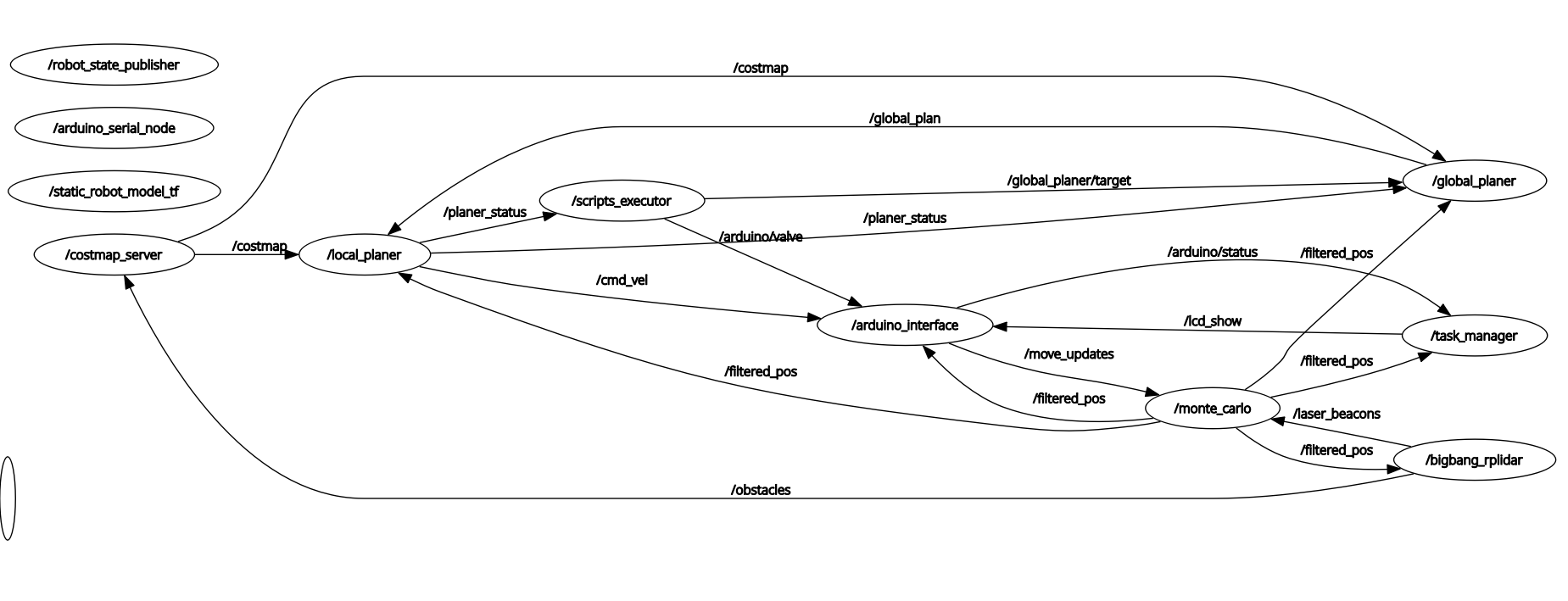
M. Fredman and R. Tarjan, “Fibonacci heaps and their uses in improved  
network optimization algorithms,” Journal of the ACM (JACM), vol. 34,  
no. 3, pp. 596–615, 1987.  
[12] E. Marder-Eppstein, E. Berger, T. Foote, B. Gerkey, and K. Konolige,  
“The office marathon: Robust navigation in an indoor office environ-  
ment,” in Robotics and Automation (ICRA), 2010 IEEE International  
Conference on. IEEE, 2010, pp. 300–307.  
[13] W. Meeussen, M. Wise, S. Glaser, S. Chitta, C. McGann, P. Mi-  
helich, E. Marder-Eppstein, M. Muja, V. Eruhimov, T. Foote et al.,  
“Autonomous door opening and plugging in with a personal robot,” in  
Robotics and Automation (ICRA), 2010 IEEE International Conference  
on. IEEE, 2010, pp. 729–736.  
[14] Towards Reliable Grasping and Manipulation in Household Environ-  
ments, New Delhi, India, 12/2010 2010.  
[15] S. Chitta, E. G. Jones, M. Ciocarlie, and K. Hsiao, “Perception,  
planning, and execution for mobile manipulation in unstructured en-  
vironments,” IEEE Robotics and Automation Magazine, Special Issue  
on Mobile Manipulation, vol. 19, 2012.

https://iopscience.iop.org/article/10.1088/1742-6596/2005/1/012095/pdf

https://habr.com/ru/articles/478836/

# Приложение А

Структурная схема навигационного стека робота BigBang



# Приложение Б

Принципиальная электрическая схема робота BigBang

# Приложение В

Электрическая схема платы расширения (драйвера моторов) Moebius

