

Министерство науки и высшего образования Российской Федерации
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технологический университет «СТАНКИН»

На правах рукописи



Ким Валерий Леонидович

МОДУЛЬНАЯ АРХИТЕКТУРА СИСТЕМЫ УПРАВЛЕНИЯ
НАЗЕМНЫХ МОБИЛЬНЫХ РОБОТОВ, ОБЕСПЕЧИВАЮЩАЯ
РЕАЛИЗАЦИЮ РАСПРЕДЕЛЁННЫХ ВЫЧИСЛЕНИЙ И
ОПЕРАТИВНУЮ РЕКОНФИГУРАЦИЮ

Специальность 05.02.05 – Роботы, мехатроника и робототехнические системы

Диссертация на соискание учёной степени
кандидата технических наук

Научный руководитель:
доктор технических наук, кандидат физико-математических наук,
профессор ФГБОУ ВО «МГТУ «СТАНКИН»
Андреев Виктор Павлович

Москва 2022 г.

ОГЛАВЛЕНИЕ

Введение	4
ГЛАВА 1. Модульные архитектуры наземных мобильных роботов.....	15
1.1 Особенности применения наземных мобильных роботов.....	15
1.2 Модульный принцип в мобильной робототехнике	20
1.3 Обзор существующих решений	23
1.4 Анализ принципов построения модульных архитектур роботов.....	38
1.5 Выводы по первой главе.....	43
ГЛАВА 2. Функционально-модульная архитектура системы управления наземного мобильного робота	45
2.1 Общее описание архитектуры системы управления	45
2.2 Принцип полной функциональности в модульной архитектуре системы управления мобильного робота	50
2.3 Информационное взаимодействие МИУ и ТМ.....	55
2.4 Информационное взаимодействие ТМ и СМБД.....	58
2.5 Выводы по второй главе.....	67
ГЛАВА 3. Система управления транспортного модуля.....	69
3.1 Модель взаимодействия полнофункциональных модулей.....	70
3.2 Динамическая модель транспортного модуля	80
3.3 Движение транспортного модуля к целевой точке.....	83
3.4 Алгоритм объезда подвижных препятствий	87
3.5 Требования к межмодульным интерфейсам ТМ и коммуникационным каналам.....	94
3.6 Реализация распределённых вычислений на основе модульной и субмодульной архитектуры СУ МР.....	100
3.7 Выводы по третьей главе.....	107
ГЛАВА 4. Межмодульное информационное взаимодействие	109
4.1 Протокол информационного взаимодействия модулей и субмодулей робота	110

4.2	Расчёт загруженности коммуникационных каналов	120
4.3	Подключение модулей в режиме «plug and play».....	126
4.4	Выводы по четвёртой главе	127
ГЛАВА 5. Моделирование и экспериментальные исследования.....		129
5.1	Компьютерная модель системы управления модульного робота	130
5.2	Исследование компьютерной модели	138
5.3	Прототипы модулей и субмодулей мобильного робота	146
5.4	Тестирование протокола ModRob	151
5.5	Экспериментальное исследование коммуникационных каналов модулей и субмодулей.....	157
5.6	Выводы по пятой главе.....	163
Заключение		165
Список литературы		168

ВВЕДЕНИЕ

Актуальность работы. В настоящее время робототехнические комплексы (РТК) активно используются не только в современных автоматизированных производствах, но и для замены человека при работе в экстремальных условиях: в космосе (на МКС), при освоении иных планет (Марс, Луна), в глубоководных миссиях (например, Марианская впадина), при работах на месте техногенных катастроф (авария на Чернобыльской АЭС и Фукусима-1 в Японии), а также для решения задач в области обороны. Во всех перечисленных случаях в составе РТК должны использоваться мобильные роботы (МР) с супервизорным или автономным управлением. В таких МР высокие требования предъявляются к системам управления, которые и обеспечивают функциональность роботов. Система управления (СУ) РТК в реальном времени должна обрабатывать значительный поток информации об окружающей среде, планировать действия, управлять исполнительными механизмами, взаимодействовать с оператором и т.д. По мере повышения требований к автономности РТК возрастает сложность алгоритмов работы систем управления, что приводит к необходимости постоянно увеличивать вычислительную мощность компьютерных систем роботов для обеспечения работы СУ в режиме *реального времени*. Согласно ГОСТ 15971-90 «Системы обработки информации. Термины и определения», режим реального времени определяется следующим образом [14]: «Режим обработки информации, при котором обеспечивается взаимодействие системы обработки информации с внешними по отношению к ней процессами в темпе, соизмеримом со скоростью протекания этих процессов». Потребность в использовании всё более и более мощных вычислительных средств существенно повышает энергопотребление РТК и, соответственно, массогабаритные параметры, что критично для автономных мобильных роботов.

В случаях использования наземных мобильных РТК в недетерминированных средах может потребоваться изменение непосредственно

на месте проведения работ состава и/или структуры РТК для точного соответствия функционала поставленной задаче. На эту проблему указывали такие известные отечественные учёные, как академик РАН д.т.н. Попов Евгений Павлович [29], д.т.н. Юревич Евгений Иванович [35] и д.ф.-м.н. Платонов Александр Константинович [27]. Например, если исследуемая наземным мобильным роботом территория крайне неоднородна по характеру рельефа и грунта, то в различные моменты времени роботу необходима как колёсная, так и гусеничная мобильная платформа. Также может возникнуть ситуация, когда должно быть увеличено количество движителей (например, колёс мобильного робота). Как показали результаты применения робототехнических средств при ликвидации аварии на Чернобыльской АЭС в 1986 г., ключом к решению такой проблемы является обеспечение возможности *реконфигурации* РТК, т.е. целенаправленного изменения конфигурации и/или технических характеристик мобильного робота при изменении состава его функциональных узлов без необходимости перенастройки программного обеспечения его СУ. Но оперативная реконфигурация РТК возможна лишь при модульном решении его конструктивной составляющей и, соответственно, программно-аппаратной части его системы управления.

Реконфигурация – изменение конфигурации модульного робота для достижения запланированного изменения функции модульного робота [86].

Само-(ре-)конфигурирование – изменение конфигурации модульного робота посредством автоматического процесса без воздействия извне системы/подсистемы, за исключением инициирования данного процесса, если это вызвано необходимостью [86]. Как правило, механические и электрические соединения должны быть сконфигурированы (или реконфигурированы) вручную, причём автоматический процесс относится к конфигурированию (реконфигурированию) *программных аспектов*.

В настоящей работе основное внимание направлено на исследование именно программных аспектов системы управления наземных РТК, включая использование вычислительных устройств, реализующих алгоритмы управления.

Требование оперативности процесса реконфигурации вытекает из экстремальных условий использования наземных мобильных РТК, когда промедления в действиях недопустимы (от этого может зависеть жизнь человека), и, к тому же, далеко не всегда реконфигурацию может выполнить человек – его должен заменить робот-ремонтник (операции в агрессивной для человека среде, в космосе, на Луне, Марсе и т.п.). Оперативность реализуется посредством создания такой системы управления модульного РТК, которая обеспечивает реконфигурацию в режиме «plug and play».

В настоящее время СУ большинства наземных мобильных роботов строятся, как правило, на базе одного вычислительного устройства значительной мощности, отвечающего сразу за все функции. Этот фактор исключает возможность оперативной реконфигурации, поскольку требует технической поддержки разработчиков РТК для перепрограммирования его СУ. Бортовой компьютер в такой системе должен обладать достаточными ресурсами, чтобы в режиме реального времени решать не только поведенческие задачи (задачи стратегического уровня), но и управлять работой всех исполнительных устройств (задачи тактического и исполнительного уровней). Для осуществления навигации и картографирования в сенсорной системе современных МР обычно используются лидары, тепловизоры и телекамеры, производящие значительный поток информации, который необходимо обрабатывать в реальном масштабе времени. Данный процесс практически невозможно реализовать с помощью одного вычислительного устройства большой производительности, энергопотребление и массогабаритные характеристики которого позволили бы устанавливать его на наземные мобильные РТК – возникает проблемная ситуация.

Обобщая сказанное, можно сделать вывод о том, что с одной стороны наземные МР должны обладать мощными бортовыми компьютерами для автономного решения поставленных задач, а с другой стороны предусматривать оперативное изменение своей конфигурации.

Решение указанных проблем видится в создании модульной архитектуры системы управления наземных мобильных робототехнических комплексов.

Модульный подход к проектированию уже зарекомендовал себя во многих отраслях промышленности: автомобилестроение, станкостроение, военная техника, компьютерная техника и т.д.

Степень разработанности темы исследования. В настоящее время модульный подход активно применяется к созданию робототехнических систем. Разработки по данному направлению ведутся в различных организациях как в нашей стране, так и за рубежом (США, Япония, Южная Корея и др.). В России проблемами модульной робототехники занимаются в ЦНИИ РТК, ИПМ им. М.В. Келдыша РАН, МГТУ им. Н.Э. Баумана, РТУ МИРЭА. Также появляются коммерческие компании, производящие наборы модулей для робототехники и автоматизации производства: Phidgets (Канада), NEBI Robotics (США), Vincross (Китай).

Существующие модульные архитектуры наземных мобильных роботов построены преимущественно на модулях, которые выполняют управление электродвигателями или опрос простейших датчиков, в то время как сложные поведенческие алгоритмы реализуются на одном бортовом компьютере. Стоит отметить, что существуют роботы, в которых модули отвечают за комплексные задачи (навигация, обработка видеоизображений). Однако проблема *оперативной* реконфигурации (и автоматической реконфигурации) системы управления наземных роботов с модульной архитектурой всё ещё остаётся актуальной задачей. Кроме того, анализ современных исследований показал, что в них предлагаются пути решения проблемы повышения автономности наземных мобильных РТК за счёт использования всё более и более производительных вычислительных систем, что противоречит требованиям снижения энергопотребления и массогабаритных характеристик мобильных систем, а также их себестоимости.

Проведённый автором обзор исследований по рассматриваемой тематике показал, что задача разработки фундаментального подхода к формализации требований к СУ роботов с позиции модульной архитектуры и методам их информационного взаимодействия является *востребованной*. Также в известных

научных работах недостаточно подробно исследуются математические или компьютерные модели наземных мобильных роботов с модульной архитектурой.

Целью диссертации является обеспечение работы в режиме реального времени системы управления наземных мобильных роботов при условиях постоянного возрастания требований к функционалу роботов и обеспечения возможности оперативной реконфигурации посредством использования методов организации распределённых вычислений¹.

Для достижения цели в работе были поставлены и решены следующие **научные задачи**:

1. Разработать функционально-модульную архитектуру системы управления наземного мобильного робота, позволяющую за счёт реализации распределённых вычислений обеспечить: а) работу СУ в режиме реального времени при условии использования вычислительных средств невысокой производительности; б) обеспечить реализацию оперативного реконфигурирования системы управления наземного МР.
2. Разработать алгоритм информационного взаимодействия колёсного транспортного модуля с модулем общесистемного управления, обеспечивающий движение робота в среде с динамическими препятствиями; определить требования к протоколу межмодульной коммуникации, межмодульным интерфейсам модуля общесистемного управления и транспортного модуля, который в свою очередь также имеет модульное исполнение; разработать (или выбрать из известных) сетевой протокол, удовлетворяющий сформулированным требованиям.
3. Выполнить компьютерное моделирование систем управления основных функциональных модулей колёсного мобильного робота, с целью доказательства реализуемости предложенной архитектуры с заданными требованиями.

¹ *Распределённые вычисления* – способ решения трудоёмких вычислительных задач с использованием нескольких компьютеров, чаще всего объединённых в параллельную вычислительную систему [32]. Распределённые вычисления применимы также в распределённых системах управления [30].

4. Провести экспериментальные исследования, подтверждающие возможность осуществления функционирования транспортного модуля в субмодульном исполнении в режиме реального времени при реализации распределённых вычислений на вычислительных устройствах невысокой производительности и использовании выбранного протокола межмодульной коммуникации.

Объект исследования – увеличение алгоритмической сложности процесса управления по мере повышения автономности наземного мобильного робота приводит к возрастанию требований к производительности вычислительных средств системы управления, что противоречит требованиям снижения энергопотребления и массогабаритных характеристик мобильных роботов, а также вызывает проблемы организации оперативного реконfigurирования системы управления.

Предмет исследования – модульная архитектура программно-аппаратной части системы управления колёсного мобильного робота.

Методы исследования. Для решения поставленных задач были использованы методы проектирования систем, основанных на коммуникациях вида «издатель-подписчик» (publisher-subscriber), метод теории конечных автоматов, методы унифицированного языка моделирования UML, методы линейной алгебры, методы распределённого и интеллектуального управления робототехническими и мехатронными системами, метод «скоростных препятствий» (Velocity Obstacle) и метод потенциальных полей для движения мобильного робота в недетерминированной среде.

Оценка соответствия алгоритмов условию исполнения в реальном времени осуществлялась путем математического моделирования, проведением экспериментальных исследований на отдельных частях лабораторного макета транспортного модуля мобильного робота.

Научная новизна результатов работы заключается в следующем:

1. Разработана новая функционально-модульная архитектура системы управления наземного мобильного робота, которая обеспечивает такую

реализацию распределённых вычислений, которая позволяет разделить общий функционал мобильного робота на более простые подфункции, что допускает их реализацию на вычислительных средствах невысокой производительности. В результате обеспечивается: а) сохранение функционирования МР в режиме реального времени при усложнении функционала МР; б) упрощается реализация оперативного реконфигурирования (режим «plug and play») системы управления МР с использованием разработок сторонних производителей.

2. Обеспечено существенное снижение трафика межмодульного информационного взаимодействия за счёт реализации в системах управления модулей и субмодулей требований принципа полной функциональности, что позволяет в системе межмодульной коммуникации использовать простые и помехоустойчивые сети промышленного стандарта CAN и CAN FD.
3. Определены необходимые и достаточные требования к коммуникационному протоколу межмодульного информационного взаимодействия в иерархической (многоуровневой) системе управления модульного МР, которые обеспечивают организацию распределённых вычислений на встраиваемых системах – микроконтроллерах и одноплатных ЭВМ невысокой производительности, что позволило выбрать из широко применяемых протоколов сетевой протокол, удовлетворяющий сформулированным требованиям.
4. Предложена математическая модель разработанной иерархической системы управления колёсного мобильного робота с функционально-модульной архитектурой, основанная на использовании метода теории конечных автоматов и диаграмм состояний UML.

Теоретическая значимость работы. Показано, что решение проблем сохранения функционирования наземного МР в реальном масштабе времени при повышении требований к автономности МР и обеспечения оперативного

реконфигурирования лежит на пути создания таких модульных архитектур систем управления, при которых функционал каждого модуля может быть разбит на подфункции, допускающие их реализацию на относительно простых вычислительных устройствах, обеспечивая тем самым иерархическую организацию распределённых вычислений.

Научные исследования в рамках диссертации выполнялись по грантам ФГБУ «Российский фонд фундаментальных исследований» (РФФИ) №16-07-00811 А (2016–2018 гг.) и №19-07-00892 А (2019–2021 гг.). Все работы по грантам были успешно завершены.

Практическая значимость работы. Следствием использования предложенных решений является снижение энергопотребления и массогабаритных параметров наземных модульных мобильных роботов за счёт применения встраиваемых систем в СУ модулей и субмодулей, а также соответствующее уменьшение стоимости разработки и ремонта роботов за счёт унификации межмодульных интерфейсов.

Требования предложенного принципа полной функциональности подразумевают, что в каждом модуле может использоваться свой набор стабилизированных напряжений, а электропитание всех модулей (включая подсистемы самого силового модуля) и субмодулей должно осуществляться от единого нестабилизированного источника напряжения МР, например, АКБ. Следовательно, должно быть реализовано распределённое стабилизированное питание модулей робота. Данное решение обеспечивает распределение выделяемой тепловой мощности между всеми модулями и субмодулями, что позволяет отказаться от установки радиатора в силовом модуле, что, в свою очередь, снижает массогабаритные параметры мобильного робота, смягчает требования к электрическим разъёмам (по току) и позволяет сторонним разработчикам использовать собственную систему стабилизированного электропитания модулей, уменьшая тем самым количество проводных межмодульных коммуникаций.

Предложенное решение СУ колёсного мобильного робота даёт возможность автоматического встраивания в СУ робота модулей сторонних производителей, что позволит постоянно расширять функционал наземных РТК без перепрограммирования остальных систем.

На защиту выносятся следующие положения:

- Показано, что сохранение функционирования наземного МР в режиме реального времени при усложнении функционала МР и упрощение организации оперативного реконфигурирования реализуется на основе предложенной новой функционально-модульной архитектуры системы управления мобильного робота, которая позволяет обеспечить такую реализацию распределённых вычислений, при которой удаётся разделить общий функционал мобильного робота на более простые подфункции, что допускает их реализацию на вычислительных средствах невысокой производительности – встраиваемых вычислительных устройствах.
- Показано, что обеспечение требований предложенного принципа полной функциональности при разработке систем управления модулей и субмодулей приводит к существенному снижению трафика межмодульного информационного взаимодействия, что позволяет в системе межмодульной коммуникации использовать простые и помехоустойчивые сети промышленного стандарта CAN и CAN FD.
- Показано, что сетевой протокол OpenCyphal удовлетворяет всем необходимым и достаточным требованиям к коммуникационному протоколу межмодульного информационного взаимодействия в иерархической (многоуровневой) системе управления модульного МР, которые определены в процессе разработки собственного протокола ModRob.
- Доказана реализуемость предложенной иерархической архитектуры системы управления на примере колёсного МР методом компьютерного

моделирования, выполненного на основе использования метода теории конечных автоматов и диаграмм состояний UML.

Степень достоверности результатов исследований. Достоверность полученных в диссертации научных результатов обеспечивается использованием методов проектирования систем, основанных на коммуникациях вида «издатель-подписчик» (publisher-subscriber), метода теории конечных автоматов, средств языка моделирования UML, применением широко известных программных библиотек и протоколов. Достоверность результатов также подтверждается непротиворечивостью результатов исследований, полученных автором, и результатов других исследователей, согласованностью теоретических выводов и положений с результатами компьютерного моделирования и экспериментальных исследований, проведённых на отдельных частях макета колёсного модульного мобильного робота.

Апробация работы. Все работы по теме диссертации проводились в рамках Грантов РФФИ 16-07-00811 А и 19-07-00892 А. Основные результаты работы докладывались и получили положительную оценку на:

- международных научно-технических конференциях «Экстремальная робототехника» (2016 г., 2017 г., 2019 г. и 2021 г., Санкт-Петербург);
- международных симпозиумах DAAAM International Symposium Intelligent Manufacturing & Automation: 2016 – Mostar, Bosnia and Herzegovina, 2017 & 2019 – Zadar, Croatia, 2020 – Mostar, Bosnia and Herzegovina (дистанционно, сертификат за лучшую статью), 2021 – Vienna, Austria (дистанционно);
- XV Юбилейной Всероссийской научно-практической конференции «Перспективные системы и задачи управления» (2020 г., п. Нижний Архыз, п. Домбай, Карачаево-Черкесской Республики);
- студенческой научно-практической конференции «Автоматизация и информационные технологии (АИТ-2016)» (2016 г., ФГБОУ ВО «МГТУ «СТАНКИН», Москва);

- международной студенческой научно-практической конференции «Автоматизация и информационные технологии (АИТ-2017)» (2017 г., ФГБОУ ВО «МГТУ «СТАНКИН», Москва);
- международной школе молодых учёных и специалистов в области робототехники, производственных технологий и автоматизации (2016 г., ЦВК «ЭКСПОЦЕНТР», ФГБОУ ВО «МГТУ «СТАНКИН», Москва).

Публикации. Результаты диссертации нашли отражение в 20 научных трудах, в том числе в 4 публикациях в изданиях из перечня ВАК РФ и в 7 статьях в изданиях, индексируемых в Scopus.

Соответствие научной специальности. Работа соответствует научной специальности 05.02.05 – «Роботы, мехатроника и робототехнические системы» в части п.2 (Математическое моделирование мехатронных и робототехнических систем, анализ их характеристик методами компьютерного моделирования, разработка новых методов управления и проектирования таких систем), п.3 (Методы адаптивного, распределенного, интеллектуального управления робототехническими и мехатронными системами) и п.5 (Методы расчета и проектирования отдельных компонентов, входящих в состав робототехнических и мехатронных систем и машин, в том числе на основе принципов оптимизации) паспорта.

Структура и объём работы. Диссертационная работа состоит из введения, пяти глав, заключения и списка литературы из 112 наименований. Работа изложена на 176 стр. машинописного текста, содержит 70 рисунков и 15 таблиц.

ГЛАВА 1. МОДУЛЬНЫЕ АРХИТЕКТУРЫ НАЗЕМНЫХ МОБИЛЬНЫХ РОБОТОВ

В первой главе диссертации рассматриваются особенности применения наземных мобильных роботов и основные проблемы разработки и эксплуатации таких роботов. Описывается модульный принцип, применяемый в мобильной робототехнике, даны основные определения, связанные с модульностью. Приведён аналитический обзор современных решений в области мобильных роботов с модульной архитектурой, который показал актуальность решения различных научных задач в этом направлении робототехники.

1.1 Особенности применения наземных мобильных роботов

В настоящее время спектр задач, для решения которых применяются мобильные роботы, значительно расширился, при этом конкретная область применения накладывает ряд специфических требований к конструкциям и системам управления роботов. Согласно ГОСТ Р 60.0.0.4-2019 [16] все роботы (в том числе мобильные) по своему функциональному назначению разделяют на две укрупнённые категории: промышленные и сервисные. Сервисные роботы в соответствии с ГОСТ Р 60.0.0.2-2016 [15] классифицируют по возможности передвижения и области применения. По области применения сервисные роботы разделяют на:

- сервисные роботы для личного и домашнего использования,
- сервисные роботы для профессионального использования.

Сервисные роботы для профессионального использования в соответствии со стандартом включают в себя более 50 видов роботов различного назначения. При этом мобильные роботы также классифицируют по среде базирования, в которой они функционируют: наземная, подводная, воздушная, космическая. В настоящей диссертации основное внимание уделяется рассмотрению вопросов, связанных с

применением *наземных* мобильных сервисных роботов, предназначенных для работы внутри помещений и на открытом воздухе, для выполнения задач наблюдения, разведки, в том числе в экстремальных условиях.

Согласно ГОСТ Р 60.6.3.1-2019 [18] *робот для работы в экстремальных условиях* (робот для аварийных работ) это «робот, предназначенный для выполнения оперативных задач в различных рабочих режимах с целью оказания помощи оператору при выполнении работ в экстремальных условиях и опасных средах». В области экстремальной робототехники широкое распространение получили колёсные и гусеничные мобильные роботы различного типа (рисунок 1.1). Конструкции шасси и СУ таких роботов постоянно совершенствуются, что позволяет использовать роботы для таких операций, как разбор завалов, разведка и обследование потенциально опасных объектов, инспектирование аварийных зон с повышенным радиационным фоном.

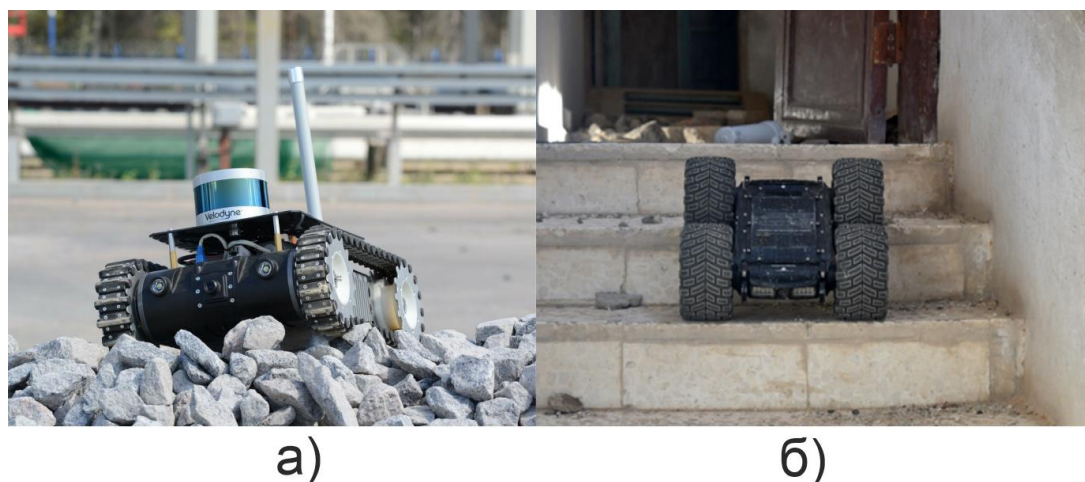


Рисунок 1.1. Малогабаритная робототехническая платформа «КУРСАНТ» (а), РФ, источник изображения [23], управляемый досмотровый робототехнический комплекс «Скарабей» (б), РФ, источник изображения [33]

Наземные МР, работающие в экстремальных условиях, можно классифицировать по массе и габаритным размерам, что является распространённой практикой, поскольку функциональные возможности МР часто коррелируют с массогабаритными характеристиками роботов. Согласно обобщённой классификации, рассмотренной в статье [12], можно выделить шесть обобщённых групп мобильных роботов в соответствии с их массой (таблица 1.1).

Таблица 1.1. Обобщённая классификация наземных МР для работы в экстремальных условиях (по материалам работы [12])

Обобщённая группа (класс)	Наименование по весу	Масса	Основное назначение
Микророботы		Менее 1 кг	Разведывательные операции, взрывотехнические работы
Мини-роботы	Сверхлёгкие мини-роботы	1–7 кг	
	Носимые мини-роботы	7–15 кг	
Сверхлёгкие		15–35 кг	
Лёгкие		35–150 кг	Разведывательные операции, взрывотехнические работы, аварийно-спасательные работы
Средние		150–800 кг	Разведывательные и военные операции, взрывотехнические работы, аварийно-спасательные работы
Тяжёлые		Более 800 кг	

В рамках диссертации интерес представляют *сверхлёгкие* наземные МР [20; 23], в частности, колёсные МР, а также *мини-роботы* с супервизорным или автономным управлением. Как показано в статье [21] роботы этих классов (с массой до 35 кг) могут перемещаться с максимальной скоростью от 10 км/ч (~ 2.78 м/с) и более; основная среда применения – здания, помещения, городские окрестности. В области экстремальной робототехники они используются для проведения разведывательных операций, картографирования, работы со взрывоопасными предметами и разминирования. Сверхлёгкие и мини-роботы используются для задач оперативного характера, поэтому их массогабаритные характеристики должны предусматривать возможность ручной транспортировки силами одного-двух человек. Обычно максимальный габаритный размер мини-роботов не превышает 700 мм, сверхлёгких роботов – не более 1000 мм [12].

Колёсные мобильные роботы существенно отличаются в зависимости от вида используемых колёс, их количества и используемой компоновки. Основные

виды колёс, используемые в наземных МР, приведены в ГОСТ Р 60.0.0.5-2019 [17] и в книге [107]. В настоящее время количество возможных конструктивных схем шасси колёсных мобильных роботов весьма велико, однако только часть из них эффективна на практике. Наибольшее распространение получили шасси с тремя, четырьмя и шестью колёсами, при этом все колёса могут быть ведущими. На рисунке 1.2 показаны популярные схемы шасси колёсных МР: трёхколёсный МР с дифференциальным приводом (а), четырёхколёсный МР с автомобильной компоновкой (б), четырёхколёсный МР с всенаправленным мобильным механизмом (в), четырёхколёсный МР с бортовым способом поворота (г) и шестиколёсный МР с бортовым способом поворота (д).

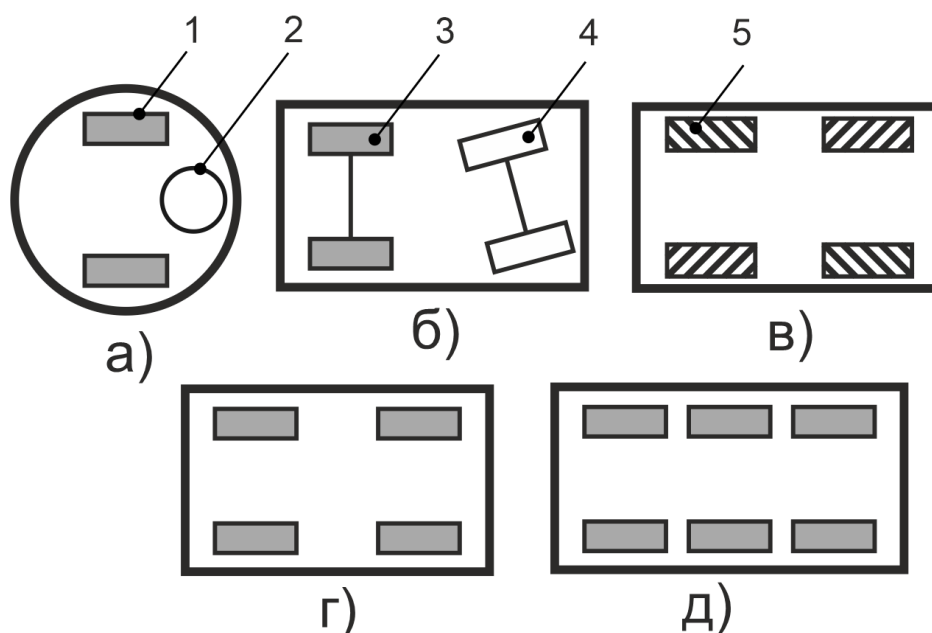


Рисунок 1.2. Широко распространённые колёсные схемы, используемые в МР.

Изображение по материалам книги [107]

Обозначения на рисунке 1.2: 1 – стандартное ведущее колесо, приводимое в движение отдельным двигателем, 2 – пассивное всенаправленное ведомое колесо (например, сферическое колесо или роликовая опора), 3 – ведущие колёса на общей оси с дифференциалом, 4 – ведомые колёса с общей осью с регулируемым поворотом, 5 – всенаправленное колесо (меканум-колесо).

Следует отметить, что не существует такой компоновки, которая была бы универсальной и соответствовала всем требованиям поставленной задачи в

условиях неопределённости окружающей среды. Поэтому востребованной является задача реконфигурирования наземных (колёсных) мобильных роботов, что обеспечивается модульным принципом построения, рассматриваемым в настоящей диссертации.

Очевидно, что особенности той или иной конструктивной схемы шасси должны учитываться в системе управления МР. Например, для МР с дифференциальным приводом или автомобильной компоновкой должны браться в расчёт неголономные связи при движении по заданной траектории или построении пути. Помимо управления приводами шасси, СУ МР должна в режиме реального времени обрабатывать показания различных датчиков, строить модель местности, планировать маршрут в среде, где есть подвижные и неподвижные препятствия различной формы и размеров.

Система управления наземного МР с супервизорным или автономным управлением должна решать следующие основные задачи навигации:

- одновременная локализация и построение карты (simultaneous localization and mapping, SLAM) [17],
- планирование маршрута и/или траектории,
- объезд препятствий и предотвращение столкновений.

Кроме того, мобильный робот должен поддерживать канал связи с оператором, а также выполнять другие функции, связанные с конкретным применением робота, например, манипулирование объектами рабочей зоны.

Выполнение всех указанных задач в режиме реального времени требует использование в составе СУ чрезвычайно мощного бортового компьютера. Особенно трудоёмкой является задача SLAM. При этом возрастающие требования к вычислительным ресурсам СУ МР противоречат требованиям к длительной автономной работе МР без подзарядки и уменьшению массогабаритных характеристик, что критично для рассматриваемого класса колёсных сверхлёгких и мини-роботов (роботы должны переноситься *человеком* на значительные расстояния).

Указанная проблема должна решаться созданием систем управления наземных МР с распределёнными вычислениями. Уже достаточно давно используются системы, состоящие из множества компьютеров (мультимикрокомпьютеры), на примере которых показана эффективность распределённых вычислений [31]. Аналогичный подход оправдан в мобильной робототехнике, поэтому в настоящее время появляется всё больше разработок, где СУ роботов построены на базе множества процессоров. Модульный принцип построения обеспечивает не только реализацию распределённых вычислений, но и позволяет создавать мобильные РТК с возможностью оперативной реконфигурации, что является важным свойством для СУ роботов, работающих в экстремальных условиях.

1.2 Модульный принцип в мобильной робототехнике

Отправной точкой развития модульного принципа в мобильной робототехнике можно считать 80-е годы XX века, когда с помощью унифицированных компонентов-модулей удалось оперативно создать группу из 15 специализированных мобильных роботов для ликвидации последствий аварии на Чернобыльской АЭС (26 апреля 1986 г.). В ходе этих работ выяснилось, что в случаях чрезвычайных ситуаций, аварий, катастроф и в агрессивных для человека средах (химическое заражение местности, высокий уровень радиации и т.п.) требуются специальные мобильные роботы, предназначенные для экстремальных ситуаций. Также оказалось, что в большинстве случаев заранее трудно предугадать вид необходимых работ, поэтому требуются роботы с переменной структурой, функционал которых можно было бы менять в полевых условиях и в процессе выполнения задания [24; 27]. Такие роботы с переменной структурой принято называть реконфигурируемыми модульными роботами [24].

По способу формирования своей структуры модульные мобильные роботы разделяют на реконфигурируемые и само-реконфигурируемые. Согласно

ISO/DIS 22166-1:2021 само-(ре-)конфигурирование (self-(re-)configuration) – это изменение конфигурации модульного робота посредством автоматического процесса без воздействия извне системы/подсистемы, за исключением инициирования данного процесса, если это вызвано необходимостью. В примечании к этому определению также отмечено, что «Как правило, механические и электрические соединения должны быть сконфигурированы (или реконфигурированы) *вручную*, причём автоматический процесс относится к конфигурированию (реконфигурированию) *программных аспектов*». Следовательно, оперативное реконфигурирование заключается в обеспечении автоматической настройки СУ МР на новую конфигурацию, что реализуется за счёт режима «plug and play».

В настоящее время проблема разработки самореконфигурируемых роботов является отдельным направлением в мобильной робототехнике. Именно за этим понятием впервые был закреплён термин «модульный робот» (Modular Robot – MR) или «модульный робототехнический комплекс» (Modular Robotic System – MRS). Поэтому в российской и зарубежной литературе, а также в научных публикациях, под модульными роботами чаще всего подразумевают именно самореконфигурируемые роботы.

Самореконфигурируемый робот (self-reconfigurable robot) представляет собой двумерный или трёхмерный массив из кинематически связанных модулей (более простых роботов) с несколькими степенями свободы, которые подсоединяются или отсоединяются друг от друга и формируют модульную структуру или конфигурацию [52]. Первый самореконфигурируемый робот был представлен Фукудой и Накагавай в 1987 году и назывался CEBOT (Cellular Robotic System) [79].

Самореконфигурируемые роботы классифицируются по многим признакам, но наиболее распространено разделение на *гомогенные* и *гетерогенные* роботы. В *гомогенных* роботах большинство модулей имеют одинаковую конструкцию и функциональность [73; 74; 95]. В *гетерогенных* роботах модули могут отличаться функциями, конструкцией, видами движений. В разных системах модули по своей

конструкции могут варьироваться достаточно широко: начиная от механических приспособлений (уголков, опор) и заканчивая более сложными устройствами.

В последнее время широкое развитие получили исследования, связанные с созданием мобильных роботов с модульной архитектурой. Они не относятся ни к гомогенным, ни к гетерогенным роботам. Роботы с модульной архитектурой следует причислять к новому направлению, которое ещё не получило чёткого определения. Гомогенные и гетерогенные роботы решают поставленную задачу с помощью изменения своей механической конфигурации – кинематической цепи, состоящей из заведомо определённого набора модулей, за счёт их перестановки и взаимного перемещения. В мобильных роботах с модульной архитектурой целевая задача решается не за счёт манипулирования блоками робота, а за счёт целенаправленной комбинации функциональных узлов-модулей и их автоматической программной интеграции в систему управления робота в целом. Это допускает расширение набора модулей за счёт устройств, разработанных сторонними производителями.

Модули в роботах с такой архитектурой могут быть совершенно различными по уровню своей сложности – начиная от электронных плат, управляющих двигателями и иными исполнительными устройствами, и заканчивая отдельными мехатронными устройствами: манипуляторами, телекамерами, мобильными платформами и т.п.

Разработчики роботов с модульной архитектурой преследуют в основном две цели:

1. Упростить работу учёных и инженеров за счёт предоставления унифицированных робототехнических компонентов, снабжённых универсальными программными интерфейсами. Это актуально в силу большого разнообразия несовместимых аппаратных и программных компонентов роботов, доступных на современном рынке.

2. Создание мобильных РТК для задач недетерминированного характера. Модульный подход позволяет создавать реконфигурируемые наземные роботы, пригодные для выполнения миссий в обстановках с непредсказуемыми

условиями; например, для применения в поисково-спасательных операциях (МЧС), при выполнении аварийно-спасательных работ.

Первая задача тесно перекликается с разработкой аппаратно-программных фреймворков (framework), межплатформенного программного обеспечения (middleware) и операционных систем (ОС) для роботов (например, ROS [103]). Фреймворки, межплатформенное ПО, операционные системы преследуют аналогичную цель – максимально ускорить разработку новых робототехнических систем из готовых компонентов, существующих и появляющихся на рынке, адаптируя их к конкретной системе.

Вторая задача связана с разработкой таких СУ, которые обеспечивают оперативное изменение функциональности наземных МР, т.е. с минимальным участием человека или даже только с помощью роботов-ремонтников. Это требует упрощения процесса интеграции модулей до уровня «plug and play».

Рассмотрим современные разработки в области модульных архитектур наземных мобильных роботов.

1.3 Обзор существующих решений

За последние 20 лет появилось большое количество исследований и разработок, так или иначе затрагивающих тему наземных мобильных роботов с модульной архитектурой [20; 22; 26; 49; 50; 53; 61–63; 72; 75; 78; 87; 93; 94; 96; 101; 104; 105; 111]. Многие работы являются только концептуальными, есть исследования, продвигающие новые стандарты проектирования или фреймворки, либо описывающие конкретные технические решения. Учитывая растущий интерес к данной области, международной организацией по стандартизации (ISO) был опубликован [86] новый стандарт «ISO/NP 22166-1:2021 Robotics – Modularity for service robots – Part 1: General requirements», который был модифицирован в 2019 году, затем в 2020 г. и 2021 г. Стандарт регламентирует многие аспекты модульной робототехники.

Модульные архитектуры наземных мобильных роботов будем рассматривать с позиции понятий и определений, которые даны в стандарте ISO 22166-1:2021(E) [86]. Модульные архитектуры обладают следующими ключевыми свойствами: системная интегрируемость (п. 4.2.3), интероперабельность (п. 4.2.4) и реконфигурируемость (п. 3.3.17).

Системная интегрируемость (integrability) – свойство системы, при котором аппаратные и программные аспекты модулей должны быть спроектированы таким образом, чтобы их можно было объединять для формирования более крупных систем с целью выполнения требуемых целевых сервисов или функций. Для обеспечения надёжного соединения модулей следует разработать соответствующие интерфейсы.

Совместимость (interoperability – интероперабельность) – модули должны быть спроектированы таким образом, чтобы их можно было легко соединять для совместной работы с другими модулями. Они должны позволять обмениваться энергией и данными через соответствующие разъёмы. Для обеспечения обмена данными должны быть определены и реализованы протоколы взаимодействия на соответствующих уровнях.

Реконфигурация (reconfiguration) – изменение конфигурации модульного робота для достижения запланированного изменения функции модульного робота. При этом *масштабирование* является элементом реконфигурирования.

В приведённом ниже анализе акцентируется внимание на принципах межмодульного взаимодействия (программные интерфейсы, формат пересылаемых данных и т.д.), степени распределения вычислений в модульных роботах и функционале модулей. При этом рассматриваются в основном колёсные мобильные роботы с массой до 35 кг (микророботы, мини-роботы и сверхлёгкие роботы).

В области модульных архитектур мобильных роботов распространены как лабораторные исследования, так и коммерческие решения. Так, в вышедшей в 2016 г. статье [53] представлена модульная мини робототехническая платформа AMiRo (рисунок 1.3), в основе которой лежит идея распределения

вычислительной нагрузки между модулями. Платформа имеет открытую программную архитектуру: содержит свободно распространяемый загрузчик OpenBLT и две открытые операционные системы для микроконтроллеров – ChibiOS и AMiRo-OS.

Стандартная конфигурация робота содержит три базовых модуля: *модуль движения* (управление движением, вычисление пройденного пути и т.п.), *модуль визуализации статуса и беспроводной связи* (организация беспроводной связи, интерфейс телекамеры), а также *силовой модуль* (энергообеспечение и базовое поведение системы). Модуль движения и модуль визуализации статуса построены на базе микроконтроллера STM32F103 с тактовой частотой 72 МГц, силовой модуль – на микроконтроллере STM32F405 с частотой 168 МГц.

Базовая конфигурация может быть расширена модулем получения изображений (содержит телекамеру), когнитивным модулем (на базе цифрового процессора DaVinci DM3730 с частотой 1 ГГц) и модулем обработки изображений (на базе программируемой пользователем вентильной матрицы (ППВМ, англ. FPGA) Xilinx Spartan 6).

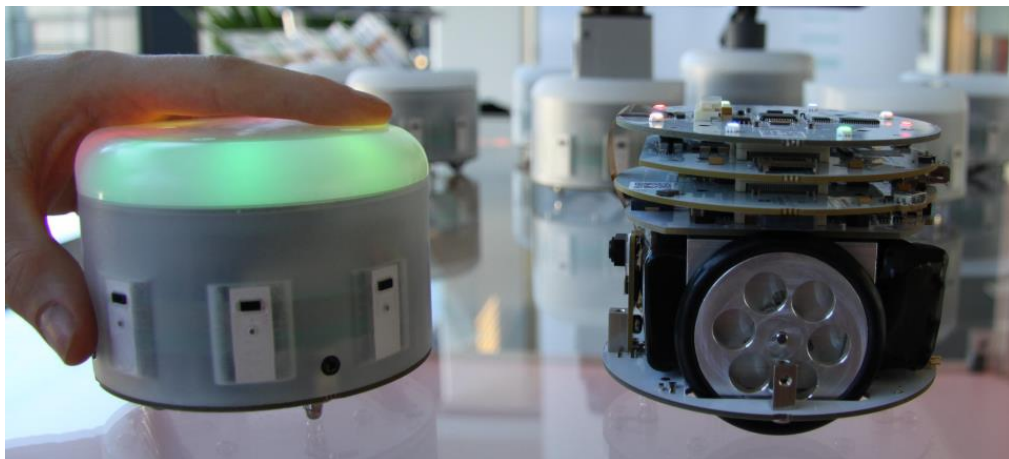


Рисунок 1.3. Базовая конфигурация AMiRo (слева) и расширенная конфигурация робота (без корпуса). Источник изображения [53]

Программная архитектура модулей построена на четырёх уровнях: загрузчик OpenBLT, операционная система реального времени ChibiOS, авторская операционная система AMiRo-OS и уровень приложения.

К достоинствам робота AMiRo можно отнести следующее:

- многофункциональность и широкие возможности для работы со сторонними приложениями,
- обеспечивается системная интегрируемость и интероперабельность между модулями AMiRo,
- работа системы в реальном времени благодаря разделению задач между модулями с мощными бортовыми компьютерами,
- поддержка шести стандартов коммуникации между модулями (CAN, UART, SPI и др.).

Недостатки системы:

- добавление в систему модулей сторонних производителей может быть затруднено из-за многоуровневой архитектуры ПО и малых габаритов робота (диаметр платформы 100 мм),
- электрические интерфейсы модулей должны поддерживать пять различных уровней напряжения питания,
- недостаточно подробно описана структура данных, используемая для межмодульной коммуникации и синхронизации,
- в статье не приводится описание экспериментов по оперативному реконфигурированию мобильного робота.

В работе 2009 года [50] представлена модульная робототехническая система DRP I, концепция которой наиболее близка к рассматриваемой в настоящем исследовании. DRP I представляет собой динамически реконфигурируемую робототехническую архитектуру, в которой каждый модуль может подключаться в режиме «plug and play».

Колёсный робот DRP I (рисунок 1.4) состоит из четырёх модулей: техническое зрение, модуль-сенсор, модуль-арбитр и транспортный модуль. Каждый модуль содержит одноплатный компьютер как основной вычислитель, а также блоки для управления модулем и тестирования.

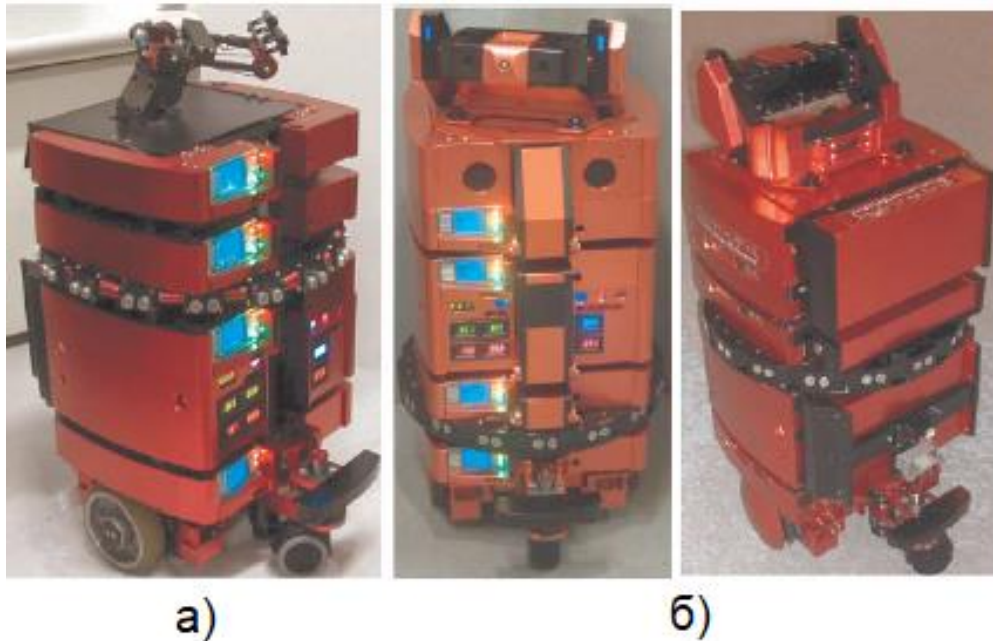


Рисунок 1.4. Модульный робот DRP I: а) старая версия, б) новая версия. Источник изображения [50]

Модули являются самостоятельными устройствами; например, транспортный модуль отвечает за перемещение робота (причём движители могут быть разными), модуль-сенсор вычисляет расстояния до препятствий и осуществляет предобработку данных, модуль технического зрения осуществляет предобработку изображений и извлекает желаемые признаки объектов. ПО каждого модуля состоит из уровня операционной системы с драйверами, уровня виртуальной машины робота (RVM), интерфейса прикладного программирования (API) и уровня прикладного ПО. Межмодульная коммуникация в роботе основана на стандарте IEEE 1394.

Авторам удалось реализовать процедуру «горячего» подключения (hot plugging) модулей к роботу. Реализована реконфигурация программного обеспечения модульного робота; например, программный код, отвечающий за режим «Объезд препятствий» может «перемещаться» между модулем-сенсором и транспортным модулем. Авторы исследования указывают, что некоторые из модулей робота были разработаны сторонними производителями.

К достоинствам робота можно отнести следующее:

- реализована системная интегрируемость и совместимость с процедурой «горячего» подключения модулей,
- реализована реконфигурируемость модульного робота,
- реализована работа в реальном времени.

Недостатки модульного робота DPR I:

- в работе используется авторская виртуальная машина RVM для модулей и авторский язык программирования RPL, что усложняет процесс интеграции модулей сторонних производителей,
- система не является полностью автономной, поскольку ни один модуль не является супервизором по отношению к остальным (необходимо участие человека-оператора).

В 2008 г. схожие по функционалу модули были реализованы в антропоморфном реконфигурируемом роботе Mom's Friend [72]. Каждый модуль может работать как независимо, так и в комбинации с другими модулями, причём их бортовые вычислители могут иметь различные операционные системы. Для обмена обработанными данными между модулями используется стек протоколов TCP/IP и стандарт RS-485.

В основе архитектуры ПО лежит «главная система» (ведущий узел), которая задаётся системой организации соединения в соответствии с назначением робота. Главная система планирует набор необходимых действий, контролирует версии других модулей. При подключении нового модуля система организации соединения принимает информацию о состоянии модуля и его характеристиках.

Для быстрого обнаружения нового модуля и получения информации о нём была разработана Архитектура Соединения (Connection Architecture) и определены стандартные информационные поля этой архитектуры. К этим полям относятся: идентификационный номер, версия, данные о концентраторе информации, IP адрес, URL адрес.

Сервисный реконфигурируемый модульный робот Mom's Friend (рисунок 1.5) состоял из 8 модулей.



Рисунок 1.5. Mom's Friend – сервисный реконфигурируемый модульный робот.

Источник изображения [72]

Рассматриваемая архитектура модульного робота не накладывает ограничения на применяемую ОС, что повышает универсальность системы и делает возможным внедрение модулей сторонних разработчиков. Кроме того, некоторые из модулей построены не на одноплатном компьютере, а на микроконтроллере (например, модуль руки).

Можно выделить следующие *достоинства* модульного робота:

- реализована системная интегрируемость и совместимость с подключением модулей в режиме «plug and play»,
- реконфигурируемость модульного робота, причём модули могут иметь различные операционные системы.
- работа в реальном времени.

В статье поверхностно раскрыт вопрос, связанный с распределением поставленной перед роботом задачи между самостоятельными модулями. Также не приводится спецификация (требования) к межмодульным интерфейсам, например, в каком формате должны пересылаться данные.

Амбициозный проект H-ROS представлен в 2017 г. в статье [111]. Цель проекта состояла в разработке универсального аппаратно-программного фреймворка для обеспечения интеграции и совместимости различных

компонентов роботов. H-ROS основан на ROS, но использует функциональность новейшей для того времени ОС для роботов – ROS 2. Модули H-ROS классифицируются в зависимости от выполняемой функции:

- получение информации о среде (телекамеры, лидары, дальномеры),
- воздействие на среду (приводы, серводвигатели),
- коммуникация (средства для организации информационного взаимодействия с другими модулями, модули, предоставляющие коммуникационные каналы – Wi-Fi, 3G, Bluetooth),
- познание-восприятие (модули для анализа поступающих данных и формирования решений, осуществления реконфигурации и др.),
- энергообеспечение (модули питания).

Физический уровень межмодульной коммуникации базируется на стандарте EtherCAT. Над физическим уровнем располагается ядро Linux с патчем (patch) реального времени PREEMPT_RT. На рисунке 1.6 показан мобильный робот Turtlebot, состоящий из H-ROS совместимых модулей.



Рисунок 1.6. H-ROS Turtlebot – первый H-ROS совместимый робот. Данный робот состоит из набора аппаратных частей, совместимых со стандартными интерфейсами H-ROS. Источник изображения [111]

К недостаткам системы можно отнести необходимость использования достаточно мощного одноплатного компьютера в каждом модуле сети (для развёртывания ОС Linux и ROS 2), что избыточно для модулей, выполняющих низкоуровневое управление исполнительными устройствами. Данное решение приводит к существенному увеличению энергопотребления, массогабаритных параметров системы, и, как следствие, – стоимости. Кроме того, для межмодульного информационного взаимодействия используется дорогостоящая технология EtherCAT. В настоящий момент проект H-ROS разработчиками остановлен [51].

В работе [87], опубликованной в 2019 г., представлена концепция модульной архитектуры роботов на базе так называемого Модуля Оператора-Контроллера (Operator-Controller Module – OCM). На базе этой концепции был разработан мобильный робот с модульной архитектурой. Архитектура разделена на локальную часть, находящуюся непосредственно на работе и осуществляющую автономное управление мобильным роботом, и облачную вычислительную среду (cloud area).

Облачная вычислительная среда, необходима для предоставления большей вычислительной мощности мобильному роботу; при этом с бортовых вычислителей робота снимается нагрузка и уменьшается энергопотребление.

Авторами был разработан экспериментальный модульный колёсный мобильный робот DAEbot с всенаправленным механизмом. Распределённые вычисления в роботе реализованы на двух одноплатных компьютерах Raspberry Pi (один компьютер отвечает за супервизорное управление другими контроллерами, а другой – за управление телекамерой), аппаратно-программной платформе Arduino Mega (опрос датчиков внутреннего состояния робота), STM32F3 Discovery (управление приводами), а также ППВМ ZynqBerry TE0726-03M (техническое зрение). Все устройства кроме ППВМ объединены CAN-шиной, протокол взаимодействия разработан самими авторами. На рисунке 1.7 представлены фотографии робота DAEbot.

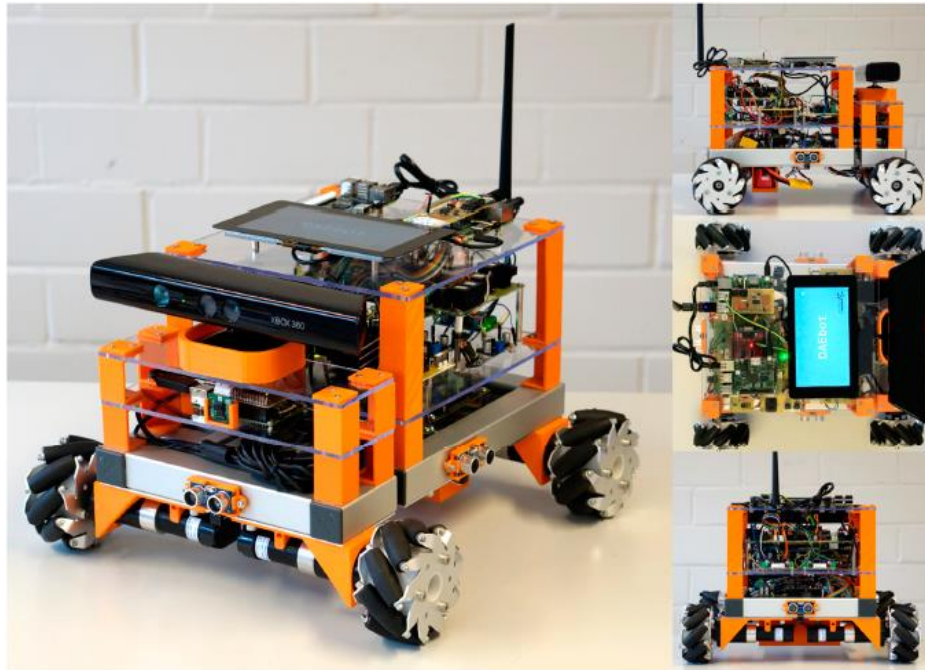


Рисунок 1.7. Экспериментальный робот DAEbot для демонстрации работы распределённой вычислительной системы на базе концепции ОСМ. Источник изображения [87]

К достоинствам данного мобильного робота можно отнести тот факт, что разработанный авторами протокол позволяет динамически настраивать частоту публикации контроллера. Это позволяет контролировать загруженность общей шины в зависимости от быстродействия подключаемых датчиков или приводов. Кроме того, благодаря мониторингу внутреннего состояния система может настраивать свое поведение в зависимости от уровня заряда батарей и температуры двигателей – переходить в энергосберегающий режим или выключать отдельные компоненты. Отметим, что в данной разработке не рассматривается унификация компонентов распределённой архитектуры, т.е. в сущности робот DAEbot состоит не из модулей, а из вычислительных устройств, объединённых CAN-шиной. По этой причине реконфигурация робота может быть существенно затруднена, поскольку в этом случае необходимо переподключение электрических соединений и перепрограммирование вычислительных устройств. Кроме того, отсутствие межмодульных интерфейсов в роботе исключает такие свойства как системная интегрируемость и совместимость. Использование

облачных вычислений приводит к существенному снижению надёжности устройства из-за использования активно работающего радиоканала.

С 2016 г. фирма Phidgets Inc. [100] продаёт наборы модулей (блоков) и программное обеспечение для создания управляющих и сенсорных систем с использованием компьютера, планшета или смартфона. Модули Phidgets подключаются в режиме «plug and play», пользователю необходимо лишь написать управляющую программу.

Компания предоставляет широкий спектр аппаратных модулей, которые можно условно разделить на два основных класса: контроллеры и сенсоры. Класс модулей-контроллеров включает в себя платы с входами/выходами, интерфейсные мосты, контроллеры электродвигателей, интерфейсы энкодеров и т.д. Класс модулей-сенсоров включает в себя модули для управления датчиками температуры, расстояния, света, движения и т.д. В основе Phidgets лежит модуль Vint Hub, служащий посредником между остальными подключаемыми модулями и компьютером разработчика. Модуль подсоединяется к порту USB компьютера. Vint Hub имеет несколько портов со специальными интерфейсами VINT.

На рисунке 1.8 показано подключение модуля-контроллера мотора (с подсоединённым двигателем) к Vint Hub.

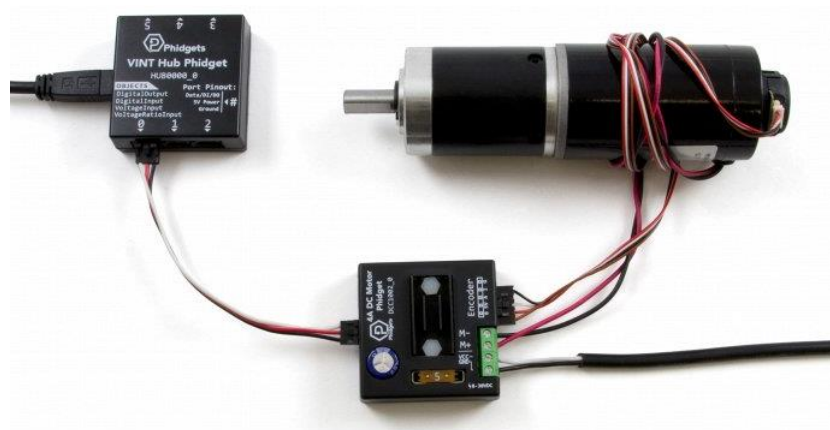


Рисунок 1.8. Контроллер двигателя постоянного тока с модулем Vint Hub.

Источник изображения [100]

Интерфейс прикладного программирования (API) Phidgets доступен для многих языков: C#, C/C++, Python, Java, Objective C и т.д. Программные

библиотеки Phidgets поддерживаются многими операционными системами: Windows, Linux, OS X, iOS и т.д.

Phidgets обеспечивает реконфигурируемость модульных систем, а также системную интегрируемость и совместимость. Однако фирма ставит перед собой иную задачу, не связанную с научными исследованиями в области модульных мобильных роботов.

Среди отечественных разработчиков следует выделить Центральный научно-исследовательский и опытно-конструкторский институт робототехники и технической кибернетики (ЦНИИ РТК), в котором уделяется особое внимание применению модульного принципа при проектировании роботов и робототехнических комплексов. Примером может служить образовательный набор «КОНСТРУКТОР» [94], малогабаритный робототехнический комплекс «КАПИТАН» [20] и унифицированная модульная архитектура СУ РТК для трёх сред базирования [22].

Образовательный набор «КОНСТРУКТОР» [94] позволяет быстро собирать различные робототехнические устройства из следующих модулей: модуль системы управления, силовой модуль, приводные модули, инструментальные модули, модули передачи данных и др. Все модули имеют стандартный электромеханический интерфейс, позволяющий осуществлять сборку роботов без применения специальных приспособлений. Коммуникация модулей осуществляется по шине RS-485.

Центральным блоком набора является модуль системы управления на базе микроконтроллера STM32F4. Модуль выполняет пользовательские программы, загружаемые через USB порт, а также поддерживает дистанционное управление с помощью встроенного Wi-Fi блока. Модуль системы управления осуществляет электропитание остальных узлов с помощью подключенного к нему силового модуля для контроля токов и напряжений. На корпусе модуля предусмотрены стандартные разъёмы PLS-3 (используются в известной аппаратно-программной платформе Arduino) для подключения устройств сторонних производителей.

На рисунке 1.9 показаны модульные роботы, собранные с помощью данного конструктора.



Рисунок 1.9. Учебные роботы, собранные из набора модулей: шестиколёсный робот (слева), манипулятор (справа). Источник изображения [25]

Несмотря на то, что набор является образовательным, организация режима «plug and play» и используемый протокол межмодульного взаимодействия представляют интерес с научной точки зрения. Однако эта информация отсутствует в описании конструктора.

Малогабаритный робототехнический комплекс «КАПИТАН» предназначен для проведения разведывательных, взрывотехнических и досмотровых операций. Комплекс состоит из универсального гусеничного шасси с изменяемой геометрией, что позволяет использовать робот в экстремальных условиях. «КАПИТАН» является реконфигурируемым мобильным РТК – на шасси допускается устанавливать различное навесное оборудование: манипулятор, тепловизор, захватное устройство, видеокамеры, комплект специализированных инструментов и т.д. Подключение навесных модулей осуществляется с помощью универсального соединителя, обеспечивающего электропитание, передачу аудио-видео потоков и управляющих данных.

На рисунке 1.10 показана одна из четырёх возможных конфигураций робота для разведки и наблюдения (навесные модули: тепловизор, видеокамера трансфокаторная и видеокамера широкоугольная). Благодаря универсальному

соединителю замена навесного оборудования осуществляется менее чем за 30 секунд.



Рисунок 1.10. Конфигурация комплекса «КАПИТАН» для разведки и наблюдения

Следует отметить, что каждым модулем робота (шасси, манипулятор, телекамера) управляет непосредственно человек-оператор с помощью джойстиков, т.е. модули комплекса функционально не взаимодействуют между собой для решения задачи (данные передаются с навесного оборудования оператору через платформу-шасси).

В статье [22] представлена унифицированная модульная архитектура системы управления РТК для трёх сред базирования: наземной, воздушной и подводной. Предложенная концепция обеспечивает системную интегрируемость РТК, совместимость модулей и реконфигурируемость. Архитектура СУ РТК включает в себя следующие основные модульные компоненты:

- центральный вычислительный модуль,
- модули полезной нагрузки (управление манипулятором, осветители),
- модуль видеокоммутации (обработка аудио и видеопотоков),
- модуль приёмо-передатчиков (обмен данными с постом управления),
- сенсорный блок,

- блок питания,
- модули для управления электродвигателями.

В статье описывается макет бортовой системы дистанционного управления, включающий в себя вышеупомянутые модули. Каждый модуль выполнен как печатная плата с унифицированным форм-фактором. Печатные платы модулей подключаются к объединяющей плате для коммутации (по аналогии с материнской платой компьютера). На рисунке 1.11 показана модель платы центрального модуля управления и модель объединяющей платы. Центральный модуль на базе микроконтроллера STM32F407VGT производит основные вычислительные операции и общее управление другими модулями РТК за счёт передачи команд по шине CAN. Также центральный модуль ответственен за формирование и передачу телеметрических данных, управление блоком реле и предохранителей.

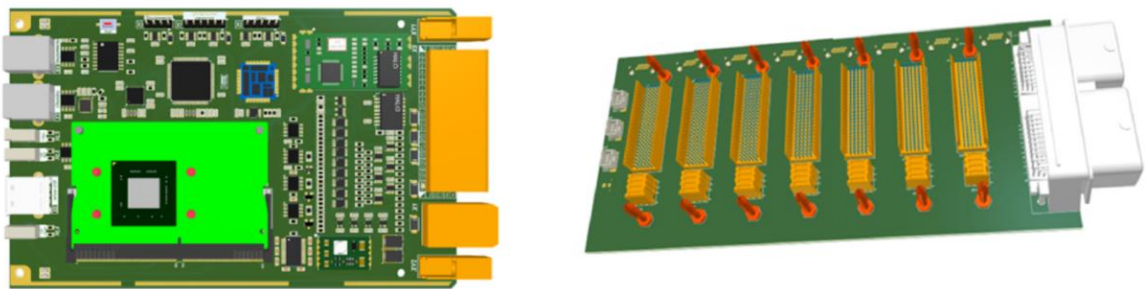


Рисунок 1.11. Модель центрального модуля управления (слева), модель объединяющей платы (справа). Источник изображения [22]

Следует отметить, что в данной работе функции модулей и вопросы их коммуникации рассматриваются поверхностно: не указывается протокол межмодульного взаимодействия по шине CAN, не описаны структуры или формат пересылаемых данных. Недостатком представленной архитектуры является загруженность центрального модуля различными задачами: передача телеметрических данных, обработка данных сенсорного блока и управление исполнительными устройствами (реле и предохранители), что требует использования высокопроизводительного бортового вычислителя. При этом в статье указано, что сенсорный блок включает в себя большой набор

разнообразных датчиков в зависимости от среды базирования (в статье отмечается универсальность модульной архитектуры).

1.4 Анализ принципов построения модульных архитектур роботов

Большая часть рассмотренных в обзоре работ посвящена созданию колёсных мобильных роботов с модульной архитектурой. В этих исследованиях показан принцип разбиения аппаратно-программной части системы управления мобильного робота на функциональные блоки, продемонстрирована реконфигурируемость и интероперабельность модулей.

Значительная часть работ посвящена программно-аппаратным фреймворкам. Фреймворк представляет собой набор аппаратных и программных средств и интерфейсов прикладного программирования (API). К фреймворку также можно отнести известную операционную систему для роботов – ROS. Фреймворки призваны облегчить труд разработчиков робототехнических устройств за счёт предоставления интероперабельных компонентов-модулей. Фреймворки также обеспечивают системную интегрируемость. Именно интенсивное развитие фреймворков способствовало появлению в 2014 и 2016 гг. таких коммерческих робототехнических систем, как Phidgets и NEBI Robotics [82].

Обзор показал, что модули во многих рассмотренных мобильных роботах выполняют низкоуровневые задачи: управление приводом в соответствии с заданным законом изменения обобщённой координаты $q(t)$ и/или скорости $\dot{q}(t)$, опрос простейших датчиков. В классической структуре систем управления интеллектуальных мобильных роботов Попова Е.П. [29] такие модули соответствуют первому исполнительному уровню (рисунок 1.12).

Рассматриваемые в отечественной литературе мехатронные модули движения (ММД) и интеллектуальные мехатронные модули (ИММ) (Подураев Ю.В. [28]) также выполняют функции исполнительного уровня. При

этом модули из обзора, соответствующие исполнительному уровню, отличаются от ММД и ИММ тем, что в общем случае контроллер управления движением не связан *механически* с конкретным двигателем, хотя есть и исключения [94].



Рисунок 1.12. Уровни управления в мехатронных системах и РТК [29].

В отдельных исследованиях встречаются мобильные роботы, модули которых решают задачи *тактического* уровня, например, перемещение в среде с объездом препятствий (робот DRP I), позиционное управление рукой робота (робот Mom's Friend). В общем случае такие модули должны формировать вектор обобщённых координат $\mathbf{q}(t) = (q_1(t), q_2(t), \dots, q_n(t))^T$ и/или вектор обобщённых скоростей $\dot{\mathbf{q}}(t) = (\dot{q}_1(t), \dot{q}_2(t), \dots, \dot{q}_n(t))^T$ звеньев модуля (n – количество звеньев, например, ведущих колёс), используя информацию о заданном положении и/или заданной скорости системы координат, связанной с мобильной платформой робота (для случая транспортного модуля – робот DRP I [50]) или со схватом руки робота (для случая модуля-манипулятора робота Mom's Friend [72]) относительно некой базовой системы координат. При этом модули тактического уровня при

необходимости должны осуществлять локальное планирование траектории с учётом кинематических и динамических ограничений и различных препятствий.

В представленных исследованиях за формирование плана движения, т.е. решение задач *стратегического* уровня, обычно отвечает человек или внешний по отношению к модульному роботу компьютер, реже – конкретный модуль-супервизор мобильного робота.

Рассмотренные исследования также можно классифицировать по степени распределения вычислений между модулями согласно предложенной в 2010 г. классификации [61]:

1. *Централизованная система управления* – один центральный процессор полностью отвечает за поведение робота, опрос датчиков и обработку их показаний, низкоуровневое управление приводами, взаимодействие с оператором и т.д. Преимущество полностью централизованной системы заключается в простоте аппаратно-программного обеспечения. Однако с ростом числа датчиков и приводов возникают следующие проблемы: загруженность общей системной шины, значительная сложность добавления новых устройств в систему, ограниченность числа входов/выходов процессора.

2. *Частично распределённая система управления* – система управления, в которой задачи исполнительного, а иногда и тактического уровня, распределены между модулями робота, которые, как правило, построены на базе микроконтроллеров. При этом обычно в системе управления есть центральный модуль с более мощным вычислителем, который координирует действия подчинённых модулей. Это наиболее распространённый тип системы управления, встречающийся в модульной робототехнике.

3. *Полностью распределённая система управления* – система построена на встраиваемых вычислителях (обычно микроконтроллерах) таким образом, что наличие центрального процессора необязательно, поскольку поведенческие функции распределяются между встраиваемыми вычислителями (модулями). Пересылка данных между модулями в полностью распределённой системе происходит только в случае возникновения какого-либо события.

Частично и полностью распределённые системы обладают следующими основными преимуществами по сравнению с централизованными:

- в целом быстродействие РТК должно быть выше из-за распределения вычислительной нагрузки между узлами,
- возможность добавления новых устройств (модулей) – «горизонтальная масштабируемость» системы,
- возможность осуществления реконфигурирования и наличие свойства интероперабельности.

Основной недостаток – распределённая система в целом менее надёжна, поскольку включает большое число модулей и периферийных устройств. Однако, это может компенсироваться простотой замены вышедших из строя модулей, либо дублированием модулей.

Организация информационного взаимодействия между вычислительными устройствами модулей порождает вторую проблему – выбор коммуникационной среды, формата сообщений, системы команд и т.д., обеспечивающих работу всех модулей робота в реальном времени. Однако, как будет показано далее, использование принципа полной функциональности при разработке модулей приводит к существенному сокращению объёма информации при межмодульном информационном взаимодействии, что значительно снижает требования к пропускной способности коммуникационных каналов.

Сравним основные характеристики рассмотренных модульных архитектур мобильных роботов (таблица 1.2). В сравнительную таблицу внесены данные о тех модульных РТК, которые близки по концепции к рассматриваемой в диссертации системе управления мобильных роботов с модульной архитектурой.

В таблице приняты следующие обозначения: Ц – централизованная система управления, ЧР – частично распределённая система управления, Р – распределённая система управления; И – модули, решающие задачи исполнительного уровня управления, Т – модули, решающие задачи тактического

уровня управления, С – модули, решающие задачи стратегического уровня управления.

Таблица 1.2. Сравнительные характеристики модульных архитектур мобильных роботов

Характеристика	AMiRo	DRP I	Mom's Friend	H-ROS	OCM	Phidgets	«КАПИТАН»	Модульная архитектура СУ РТК
Ссылка	[53]	[50]	[72]	[111]	[87]	[100]	[20]	[22]
Степень распределения вычислений в роботе	ЧР	ЧР, Р	ЧР	ЧР	ЧР	ЧР	ЧР	ЧР
Основные виды модулей в роботе	Т	Т	Т, С	И, Т	И, Т	И	И, Т	И, Т
Стандарт межмодульной коммуникации	CAN, UART	IEEE 1394	TCP/IP	EtherCAT	CAN	USB, VINT	н/д	CAN
Работа в реальном времени	+	+	+	+	+	+	+	+
Наличие операционной системы на модулях	+	+	+	+	+	н/д	н/д	н/д

Из таблицы видно, что большинство рассмотренных модульных РТК построены как частично распределённые системы. Несмотря на то, что в таблице основные виды модулей решают задачи тактического уровня управления, обстоятельный аналитический обзор, проведённый автором, показывает, что в большинстве известных модульных роботов системы управления построены на базе модулей исполнительного уровня. Поэтому большой интерес представляет вопрос реконфигурации и обеспечения работы в реальном времени мобильных РТК, состоящих из модулей, которые решают задачи стратегического и тактического уровней управления.

В представленных модульных архитектурах вопрос межмодульных программных интерфейсов рассматривается поверхностно, т.е. не указывается,

какие данные пересылаются между модулями, в каком формате, с какой частотой и т.д. Иначе говоря, нет фундаментального подхода к формализации требований (спецификации) к СУ с позиции модульной архитектуры и методам их информационного взаимодействия. Это связано с тем, что в рассмотренных исследованиях решаются сугубо прикладные задачи, соответственно недостаточно подробно описываются математические или компьютерные модели мобильных роботов с модульной архитектурой. Обзор также показал, что недостаточно внимания уделяется вопросу, связанному с решением проблемы обеспечения режима реального времени при повышении требований к автономности модульных наземных мобильных роботов. Предлагаются лишь частные решения, основанные, преимущественно, на повышении производительности вычислительных устройств в СУ модулей.

Принципы распределения задач между модулями также почти не раскрываются. В то же время быстродействие модульной системы в значительной степени зависит от способа распределения функций. Формализация этой задачи должна проводиться с учётом различных сценариев работы модульного робота.

1.5 Выводы по первой главе

Для наземных (в частности, колёсных) мобильных роботов, функционирующих в недетерминированных условиях в автономном режиме, задача реконфигурации шасси роботов и распределение вычислений между отдельными процессорами *является актуальной*. Решение этой задачи должно быть основано на использовании модульного принципа построения систем управления наземных МР.

Аналитический обзор показал, что существует достаточно большое число исследований и разработок в области модульной мобильной робототехники, выполненных преимущественно за рубежом. При этом остаётся востребованным решение следующих проблем:

1. Развитие модульных архитектур для решения проблемы обеспечения режима реального времени при повышении требований к автономности наземных мобильных роботов.
2. Формализация требований к межмодульным информационным интерфейсам: структуры пересылаемых данных, формат данных, частота передачи и т.д.
3. Реконфигурация и обеспечение работы СУ в реальном времени для мобильных роботов, состоящих из модулей, которые решают задачи стратегического и тактического уровней управления.
4. Исследование принципов распределения функций между модулями, обоснование принятого функционала модулей.
5. Исследование математических и компьютерных моделей СУ мобильных роботов с модульной архитектурой, в которых учитывается работа СУ и информационное взаимодействие модулей в режиме реального времени.

Решения рассматриваемых проблем приведены в следующих главах настоящей диссертации.

ГЛАВА 2. ФУНКЦИОНАЛЬНО-МОДУЛЬНАЯ АРХИТЕКТУРА СИСТЕМЫ УПРАВЛЕНИЯ НАЗЕМНОГО МОБИЛЬНОГО РОБОТА

В главе описывается предлагаемая иерархическая функционально-модульная двухуровневая архитектура системы управления колёсного мобильного робота [54]. В этой архитектуре используется разработанный автором принцип полной функциональности [57], позволяющий снизить вычислительную нагрузку, приходящуюся на каждый модуль робота, и сократить объём трафика при межмодульном информационном взаимодействии.

В главе приведён анализ информационного взаимодействия полнофункциональных модулей и определены структуры пересылаемых данных и их формат между модулем общесистемного управления и транспортным модулем, а также между транспортным модулем и сенсорным модулем ближнего действия [56].

2.1 Общее описание архитектуры системы управления

В работах [5; 10; 54; 60] автором (совместно с коллегами) было предложено создавать систему управления, реализующую распределённые вычисления на множестве процессоров по аналогии с мультиагентными системами. В этой СУ общая задача разбивается на подзадачи по функционально-модульному принципу при распределении подзадач между модулями [57]. Полагаем, что такой подход позволит существенно снизить вычислительную нагрузку на вычислительное устройство каждого модуля, что даст возможность использовать в системах управления модулей встраиваемые системы – относительно маломощные микроконтроллеры или одноплатные ЭВМ. Это даст возможность снизить энергопотребление каждого модуля, что в свою очередь позволит отказаться от радиаторов на процессорах и уменьшить массогабаритные характеристики устройств. Как следствие – уменьшается стоимость производства модульных МР.

Принцип разбиения структуры мобильного робота на функциональные модули может быть различным в зависимости от поставленной задачи. В то же время известно, что практически в любом мобильном роботе с элементами интеллектуального или адаптивного управления в той или иной степени должны быть представлены следующие функции [19; 34; 102; 107]:

- общесистемное управление (интеллектуальная функция),
- информационная функция,
- технологическая функция,
- транспортная функция,
- энергетическая функция,
- коммуникационная функция.

Например, в [34] рассматривается обобщённая структурная схема системы управления «очувствлённым» роботом, которая включает в себя уровень общего функционирования (*общесистемное управление*), сенсорные устройства (*информационная функция*), технологические операции (*технологическая функция*), управление приводами (*транспортная и технологическая функция*), а также пульт управления оператора (*коммуникационная функция*). Те или иные функции в конкретном роботе могут быть более-менее развиты или отсутствовать вовсе.

Предлагается поставить в соответствие каждой функции свой аппаратно-программный модуль, имеющий собственную систему управления. Функционал каждого модуля должен определяться исходя из предложенного автором принципа полной функциональности [4; 57]: *каждый модуль робота должен быть способен любым удобным ему способом выполнять свою целевую функцию, используя только собственные средства для выполнения команд от внешней системы управления*. Тогда существенно снижается межмодульный трафик, что даёт возможность использовать помехозащищённые каналы с невысокой пропускной способностью типа шины CAN, широко распространённой в современных транспортных средствах.

На основании анализа результатов проведённого обзора и предложенного принципа полной функциональности была разработана следующая структура иерархической модульной двухуровневой архитектуры системы управления наземного мобильного робота (рисунок 2.1) [6].

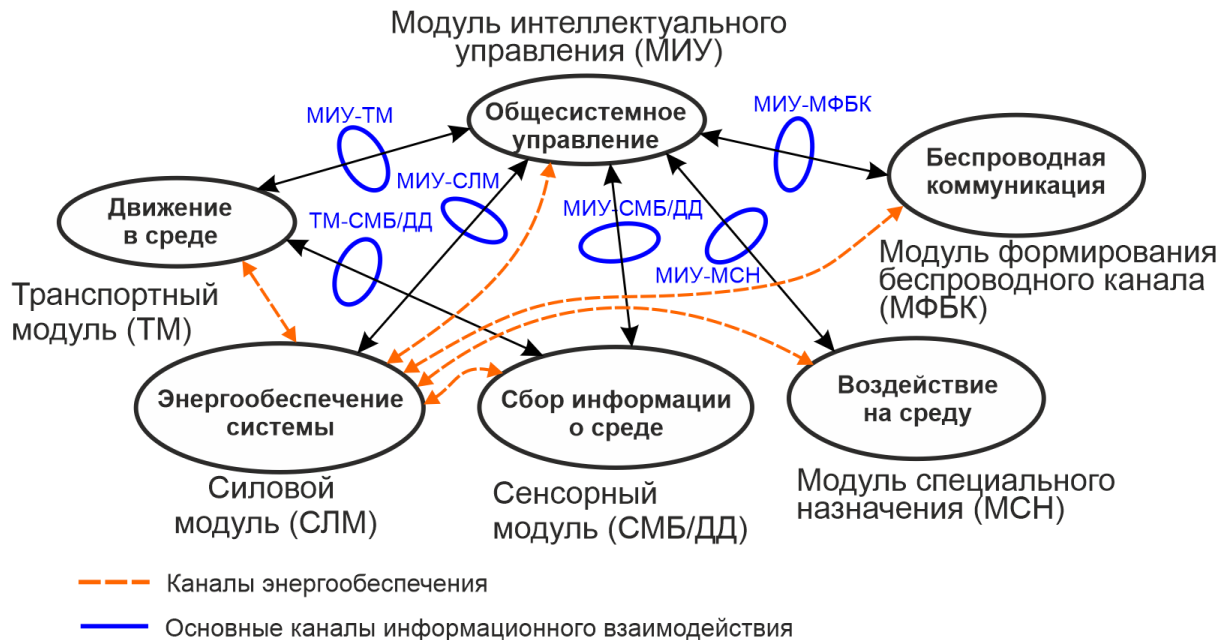


Рисунок 2.1. Обобщённая иерархическая модульная двухуровневая архитектура системы управления наземного мобильного робота

На основе принципа полной функциональности предложено рассматривать структуру СУ МР (в минимальном исполнении) как синергетическое объединение следующих полнофункциональных модулей [6; 54]: модуль интеллектуального управления (МИУ), транспортный модуль (ТМ), силовой модуль (СЛМ), сенсорный модуль (СМ), состоящий из сенсорного модуля ближнего действия (СМБД) и модуля дальнего действия (СМДД), модули для активного взаимодействия с внешней средой – модули специального назначения (МСН): манипуляторы, схваты и т.д. В этой структуре каждый модуль отвечает за одну укрупнённую функцию целого мобильного робота.

Общесистемное управление. Функция реализуется в отдельном модуле МИУ, который является супервизором по отношению к другим модулям робота. В данной архитектуре модуль-супервизор формирует цель управления и контролирует лишь результат её достижения модулем-исполнителем, а не

управляет процессом выполнения задачи этим модулем; это обусловлено принципом полной функциональности. Такая функциональная завершённость является основным отличием нашего решения от большинства работ рассмотренных ранее, в которых модуль-супервизор управляет работой всех модулей на *исполнительном уровне*. Таким образом, МИУ отвечает за планирование и распределение задач между модулями второго уровня.

Транспортная функция реализуется *транспортным модулем* (ТМ) – модуль представляет собой колёсную мобильную платформу, которая обеспечивает перемещение робота в пространстве. Полная функциональность модуля заключается в следующем. ТМ может иметь различное конструктивное исполнение в зависимости от вида используемых колёс, их количества и компоновки (рисунок 2.2). Его задача переместить модульный робот из текущего положения в целевое, заданное модулем МИУ. Алгоритм управления перемещением ТМ будет различаться в зависимости от конкретной колёсной схемы, и должен реализовываться собственной системой управления ТМ. В случае необходимости ТМ может получать информацию о среде от сенсорных модулей робота.

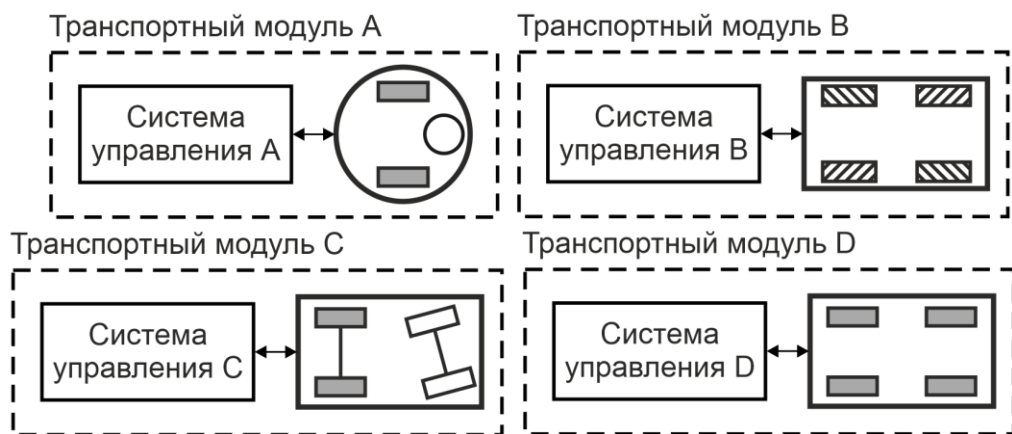


Рисунок 2.2. Различные варианты транспортного модуля

Информационная функция реализуется дистанционной сенсорной системой, состоящей из нескольких сенсорных модулей: *модуля датчиков ближнего* (СМБД) и *модуля датчиков дальнего действия* (СМДД). Модули решают задачу SLAM, обнаруживают препятствия и объекты манипулирования и предоставляют

сенсорную информацию другим модулям (по запросу или с заданным периодом). Отметим, что каждый модуль также снабжается собственной сенсорной системой, состоящей из датчиков внутреннего состояния.

Технологическая функция (и частично транспортная функция) – воздействие на среду реализуется модулями специального назначения (МСН). Это модули манипуляторов-актуаторов и модули захвата, которые дают возможность использовать различную технологическую оснастку для взаимодействия с объектами манипулирования в пределах рабочей зоны робота. К реализации технологической функции также относятся модули механических операций – технологическая оснастка для модуля манипуляторов, позволяющая совершать различные технологические взаимодействия со средой – сверление, фрезерование и т.п.

Коммуникационная функция реализуется за счёт сетевой организации информационного взаимодействия между модулями (на рисунке 2.1 обозначена разнонаправленными стрелками). Коммуникация МР с внешним супервизором – оператором робототехнического комплекса формируется специальным модулем формирования беспроводного канала связи (МФБК). На схеме приведена топология сети типа «звезда», но возможна и «шинная» топология, а также смешанная.

Энергетическая функция реализуется в силовом модуле (СЛМ) – модуль обеспечивает электропитание электронных и электромеханических компонент модулей робота, а также (согласно принципу полной функциональности) их безопасное включение и выключение: завершение работы программных модулей, сохранение состояния файловой системы. Силовой модуль должен отслеживать уровень заряда аккумуляторных батарей (АКБ) и сообщать о низком уровне заряда в МИУ. Отключение электропитания того или иного модуля выполняется лишь по команде МИУ – всех модулей (включая себя) в случае завершения работы, или тех модулей, которые не нужны для выполнения работ на текущий момент (например, для экономии энергии).

На основе силового модуля реализуется принцип распределённого стабилизированного питания [5]. Его особенность заключается в том, что общее нестабилизированное питание всех модулей робота и исполнительных устройств (приводов) осуществляется от единого аккумуляторного блока, входящего в состав СЛМ. При этом в каждый модуль устанавливается блок формирования требуемого набора стабильных напряжений постоянного тока (БФСН), обеспечивающий стабилизированным напряжением внутримодульные электронные компоненты, например, микроконтроллеры. Данное решение позволяет уменьшить количество проводных межмодульных коммуникаций, а также становится возможным организовать информационный межмодульный канал по проводным линиям электропитания [1].

На вход БФСН подаётся напряжение непосредственно от АКБ, но под управлением СЛМ. При пульсациях входного напряжения, вызванных, например, пусковыми токами приводов, наличие БФСН исключает возможную перезагрузку микроконтроллеров.

В случае превышения суммарной потребляемой мощности, достаточно лишь установить в СЛМ дополнительные АКБ или заменить сам силовой модуль. Кроме того, тепло, выделяемое в процессе стабилизации напряжений, распределяется между модулями, а не концентрируется в едином блоке питания, что позволяет отказаться от установки габаритного радиатора в СЛМ.

2.2 Принцип полной функциональности в модульной архитектуре системы управления мобильного робота

В данной диссертации основное внимание уделяется транспортному модулю и его взаимодействию с сенсорным модулем ближнего действия и модулем интеллектуального (общесистемного) управления, поскольку это ключевые элементы мобильного робота. Рассмотрим *принцип полной функциональности* на примере транспортного модуля [4].

Пусть модульный робот перемещается на плоской подстилающей поверхности в некоторой среде. В окружении робота могут находиться статические и динамические (подвижные) препятствия (например, люди, другие роботы). Модульный робот получает общее задание от внешнего-супервизора (обычно это человек-оператор). Общее задание включает в себя задачу перемещения в целевую позицию, которую можно описать вектором $[X_G, Y_G]^T$, где X_G, Y_G – координаты целевой позиции в некоторой базовой системе координат. В рассматриваемой задаче угол ориентации в целевой позиции не важен.

МИУ, ТМ и СМБД (опционально СМДД) решают задачу перемещения совместно, но каждый из них отвечает только за свою функцию (таблица 2.1).

Таблица 2.1. Основные функции модулей

Модуль	Основные функции	Соответствие уровням классической структуры системы управления Е.П. Попова (рисунок 1.12)
МИУ	Глобальное планирование и формирование задач для других модулей	Стратегический, частично интеллектуальный
ТМ	Локальное планирование и перемещение	Тактический, исполнительный
СМБД	Сбор и обработка информации об окружающей среде в ближней зоне	Выдача сенсорной информации для всех уровней кроме исполнительного
СМДД	Сбор и обработка информации об окружающей среде в дальней зоне	

В настоящей работе принимается, что сенсорная система СМБД состоит из датчиков, которые используют активную локацию для обнаружения подвижных и неподвижных препятствий в плоскости. К таким датчикам относятся ультразвуковые и лазерные дальномеры, лидары (Light Detection and Ranging - LiDAR), которые сканируют пространство в одной плоскости (2D LiDAR).

Основные допущения при решении задачи:

- маршрут ТМ необязательно должен быть оптимальным по длине пройденного пути и траектории (но желательно близким к оптимальному);
- должно быть задано предельное время прибытия ТМ в требуемую позицию (дедлайн);
- по возможности ТМ должен объезжать все препятствия (в том числе подвижные); в случае столкновения с препятствием ТМ должен выполнить алгоритм останова, сообщить об этом в МИУ и ждать дальнейших указаний;
- все препятствия являются твёрдыми телами с габаритами, позволяющими их обнаруживать сенсорной системой;
- все препятствия попадают в плоскость сканирования дальномеров СМБД;
- количество подвижных препятствий в поле зрения СМБД не должно превышать некоторой критической величины.

На рисунке 2.3 показана общая схема информационного взаимодействия полнофункциональных модулей: МИУ, ТМ, СМБД и СМДД.

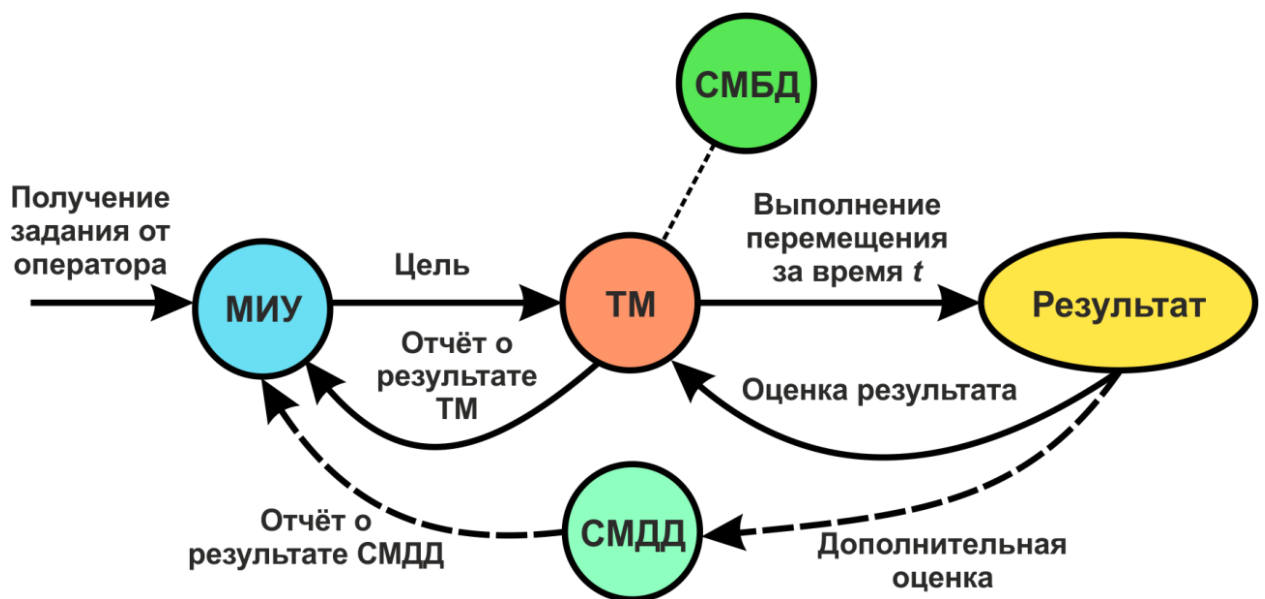


Рисунок 2.3. Общая схема информационного взаимодействия основных полнофункциональных модулей

МИУ получает от внешнего супервизора общее задание, которое он разделяет на задачи для модулей. Такой задачей для ТМ является перемещение в целевую позицию с координатами (X_i, Y_i) с некоторой допустимой ошибкой по положению. В общем случае эта целевая позиция является одной из многих точек спланированного МИУ маршрута, конечная позиция которого имеет координаты X_G, Y_G .

Согласно принципу полной функциональности ТМ должен самостоятельно достичь заданной целевой позиции, используя методы управления, заложенные в его системе управления, и информацию, получаемую от своей сенсорной системы. Предполагается, что сенсорная система ТМ должна обладать минимальным функционалом, достаточным лишь для объезда статических препятствий. Для объезда динамических препятствий и статических препятствий сложной формы должны использоваться данные, получаемые непосредственно от СМБД. Для этого в иерархической архитектуре СУ требуется наличие горизонтальных коммуникационных каналов – между модулями одного уровня иерархии.

После того, как ТМ получил задание, он выполняет перемещение к целевой позиции, одновременно используя сенсорные данные СМБД. Здесь возможны следующие ситуации:

1. Транспортный модуль заканчивает перемещение (считает, что выполнил задание по «мнению» его СУ), производит оценку своего текущего положения и отправляет в МИУ отчёт о результате. МИУ с помощью сенсорной системы МР и соответствующих алгоритмов проверяет: если положение робота находится в допустимых пределах от заданной позиции, то *задача выполнена*.
2. Транспортный модуль заканчивает перемещение (считает, что выполнил задание по «мнению» его СУ), производит оценку своего текущего положения и отправляет в МИУ отчёт о результате. МИУ проверяет: если положение робота находится за допустимыми пределами от заданной позиции, то *задача не выполнена*.

3. Аварийный останов – ТМ прекращает движение. Это может возникнуть из-за столкновения с препятствием или отказом каких-либо внутренних систем модуля. Также ТМ должен остановиться, если истекло время прибытия к цели. ТМ должен отправить в МИУ отчёт о текущей ситуации.

МИУ использует отчёт ТМ о результатах выполнения задачи (а также отчёты других модулей), чтобы сформировать новые цели или доложить человеку-оператору об успешно выполненном общем задании. Следует отметить, что оценка ТМ результата своей деятельности не всегда может быть верной, поскольку множество факторов оказывают влияние на выполнение точного перемещения робота (погрешности датчиков СМБД, проскальзывания колёс ТМ, неровность подстилающей поверхности и т.д.). Из-за этого СУ ТМ может считать, что целевое положение достигнуто, хотя это не так, или наоборот. Поэтому в данной системе МИУ всегда проверяет результат работы ТМ.

Для получения более точной оценки действительности МИУ может использовать показания СМДД, использующего такие датчики, как телекамеры, системы ГЛОНАСС, лазерные дальномеры с большим радиусом действия и т.п. На рисунке 2.3 контур управления с СМДД обозначен штриховыми стрелками, поскольку подключение СМДД для дополнительной оценки результата в исследуемом вопросе является *опциональным*.

В рассмотренной схеме взаимодействия ТМ может быть снабжён колёсным движителем с различной компоновкой колёс, но он должен отрабатывать заданную цель независимо от своей механической конструкции, структуры системы управления и алгоритма управления (см. рисунок 2.2). Кроме того, если два ТМ имеют идентичную конструкцию, их алгоритмы управления также могут отличаться при условии, что заданная цель достигается. Например, в одном ТМ используются цифровые регуляторы с нечёткой логикой, а в другом – классические цифровые пропорционально-интегральные регуляторы.

В то же время, система управления ТМ должна соответствовать интерфейсам межмодульного взаимодействия, иначе модуль не может являться частью робота, и реконфигурация неосуществима.

2.3 Информационное взаимодействие МИУ и ТМ

Модуль интеллектуального управления осуществляет глобальное планирование движения на плоскости, как было отмечено выше. МИУ формирует маршрут для транспортного модуля из исходного положения в целевое, используя данные об окружающей обстановке, представленные в виде карты местности. Обычно карта местности содержит информацию о статических препятствиях. Данные о динамических препятствиях поступают от сенсорных модулей в процессе движения робота, при этом происходит обновление карты местности.

Существуют различные методы глобального планирования пути на карте для поиска оптимального маршрута. К ним можно отнести известные алгоритмы поиска на графе: A^* (А-звезда) [81], динамический A^* [108], FD^* [109], алгоритмы случайной выборки: RRT [89], RRT* [88] и др. Выбор конкретного алгоритма поиска зависит от формального описания карты местности.

В результате планирования маршрута МИУ должен сформировать набор подцелей-точек, которые будут отрабатываться ТМ. При этом, как было отмечено ранее, путь ТМ к очередной точке необязательно должен быть оптимальным.

Пусть модулем интеллектуального управления построен маршрут из исходного положения в конечное целевое. Способ построения маршрута может быть любым, при этом маршрут должен представлять собой конечный набор точек $\{(X_i, Y_i)\}$, $i = 1, \dots, n$. Расстояния между точками должны быть не меньше некоторой величины (зависящей от размеров робота, например, не менее 3 м), иначе взаимодействие между МИУ и ТМ будет приближаться к *непрерывному*, что сводит на нет рассмотренные выше преимущества принципа полной

функциональности. Выбор минимального расстояния между точками будет рассмотрен ниже.

Иллюстрация движения модульного мобильного робота, состоящего из МИУ, ТМ и СМБД представлена на рисунке 2.4.

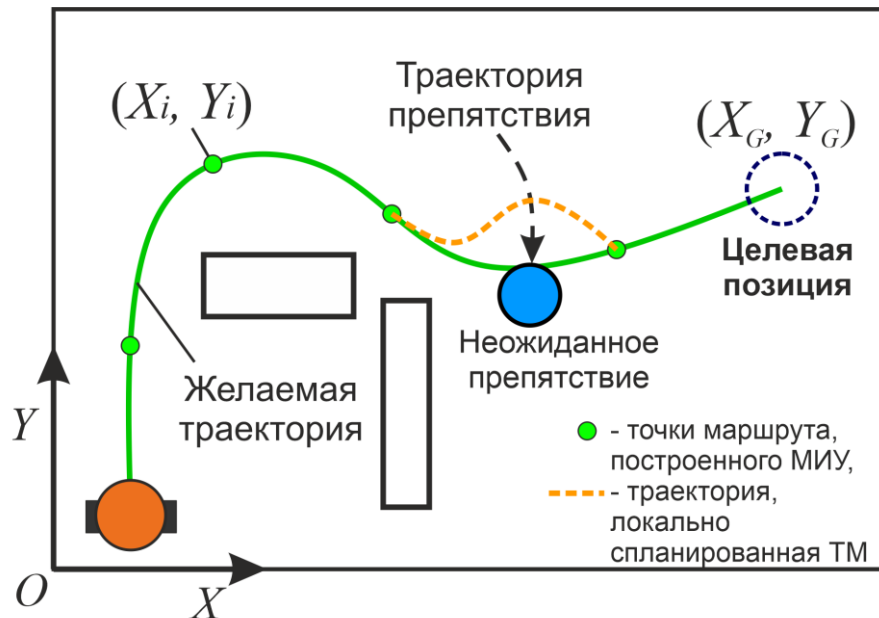


Рисунок 2.4. Иллюстрация движения модульного робота в известной для МИУ среде с подвижным препятствием

На иллюстрации показано, что в случае возникновения подвижного препятствия ТМ должен попытаться самостоятельно объехать его и добраться до очередной целевой точки, используя данные СМБД и свой локальный планировщик движения. При этом во время движения может произойти аварийная ситуация: столкновение (или высокая вероятность столкновения) с препятствием. ТМ должен распознать такую ситуацию, выполнить аварийный останов и сообщить об этом в МИУ (отправить отчёт). Кроме того, ТМ должен выполнять периодическую диагностику своих подсистем. Если произошёл отказ какого-либо узла, ТМ должен немедленно остановиться и сообщить об этом в МИУ.

Когда СУ ТМ фиксирует выполнение задания (прибытие в целевую точку), ТМ должен выполнить штатный останов и отправить отчёт в МИУ. Формальное описание взаимодействия МИУ и ТМ приводится в главе 3.

Задача на перемещение, отправляемая МИУ в ТМ, представляется следующей структурой данных:

Таблица 2.2. Структура данных задания, отправляемая МИУ в ТМ

Поле данных	Тип данных
Координата X_i целевой точки, м	Число с плавающей запятой
Координата Y_i целевой точки, м	Число с плавающей запятой
Координата X_0 текущей исходной точки, в которой находится ТМ (робот), м	Число с плавающей запятой
Координата Y_0 текущей исходной точки, в которой находится ТМ (робот), м	Число с плавающей запятой
Допустимое отклонение от целевой точки (радиус ε_{don}), м	Число с плавающей запятой
Предельное время прибытия $t_{\max}$ (дедлайн), с	Положительное целое число

Отчёт о выполненной работе ТМ отправляет в МИУ в трёх случаях: произошла аварийная ситуация, задание выполнено, ТМ неисправен. Отчёт представляется в виде следующей структуры данных:

Таблица 2.3. Структура данных отчёта, отправляемая ТМ в МИУ после выполнения перемещения

Поле данных	Тип данных
Координата X_r текущей позиции, м	Число с плавающей запятой
Координата Y_r текущей позиции, м	Число с плавающей запятой
Собственная оценка выполнения задачи	Положительное целое число

В таблице 2.3 поле «Собственная оценка выполнения задачи» отражает результат работы ТМ:

- очередная цель успешно достигнута,
- аварийный останов,
- внутренняя неисправность ТМ.

Аварийный останов может произойти в силу нескольких причин: столкновение с препятствием, высокая вероятность столкновения, истекло время прибытия в очередную точку.

Следует отметить, что целевая и текущая позиции робота задаются относительно *базовой системы координат*, связанной, например, с помещением. При этом МИУ знает расположение неподвижной базовой системы координат и системы координат самого робота.

2.4 Информационное взаимодействие ТМ и СМБД

Выполнение перемещения транспортного модуля к цели и объезд препятствий осуществляются при получении информации от сенсорного модуля ближнего действия. Эта информация должна позволить ТМ выполнить необходимые манёвры для объезда как статических, так и динамических (подвижных) препятствий.

Следует отметить, что согласно принципу полной функциональности, разделение вычислительной нагрузки между СМБД и ТМ при выполнении объезда препятствий можно реализовать множеством способов. Например, СМБД может сканировать среду в ближней зоне, находить препятствия и выдавать вектор направления движения для ТМ, который в свою очередь должен отрабатывать только перемещение в этом направлении. С другой стороны, СМБД может отвечать только за сканирование и периодически отправлять в ТМ массив точек для дальнейшей обработки.

В обоих случаях присутствует неравномерность вычислительной нагрузки, приходящейся на модули. Проблема распределения нагрузки между этими полнофункциональными модулями требует отдельного исследования, поскольку принцип такого распределения зависит от многих параметров:

- производительность вычислителей ТМ и СМБД,
- способ представления карты местности,

- сложность алгоритмов работы СУ ТМ и СМБД,
- количество уровней иерархии субмодульной архитектуры СМБД и т.д.

Следует учитывать также и то, что у процессора ТМ есть и другие задачи – управление исполнительными приводами. В первом приближении следует разделить задачу объезда препятствий так, чтобы вычислительная нагрузка между СМБД и ТМ делилась примерно поровну.

Для решения этой проблемы были рассмотрены различные алгоритмы объезда подвижных и неподвижных препятствий [13; 64; 65; 67–71; 76; 77; 90; 91; 97; 98; 106; 110], предназначенные, в основном, для плоской среды. Для данной работы интерес представляют следующие характеристики алгоритмов:

- необходимость в глобальной карте местности,
- используемый метод планирования пути,
- форма представления локальной среды,
- форма представления робота и препятствий,
- используемая кинематическая модель робота,
- учёт динамических характеристик робота.

Обзор показал, что небольшое количество алгоритмов используют глобальную карту местности, поскольку в изменяющейся среде с динамическими препятствиями важно локальное планирование для оперативного принятия решений – необходимо применять реактивную навигацию.

Планировщик пути используется в тех алгоритмах, где необходимо построение оптимального маршрута на глобальной сетчатой карте занятости или графе состояний. В качестве планировщика применяют алгоритм A^* и различные методы оптимизации.

Локальная среда вокруг робота, включающая в себя подвижные и неподвижные препятствия, может представляться различными способами. Например, в методе скоростных препятствий (Velocity Obstacle) [76] локальная среда представляется как область достижимых полей скоростей робота, строящаяся на базе его ускорений и «скоростных конусов»

подвижных/неподвижных препятствий. В работах [13; 68] локальная среда представляется как сетчатая карта занятости (occupancy grid), что удобно для локализации как статических, так и динамических препятствий. В известном методе VFH (Vector Field Histogram [65]) точки статических препятствий локальной среды, полученные после сканирования пространства ультразвуковыми датчиками, отображаются на сетчатую карту занятости (histogram grid), которая затем преобразуется в полярную гистограмму. В методе Nearness Diagram [91] множество точек препятствий используется для построения круговых диаграмм, разбитых на сектора. При этом каждому сектору ставится в соответствие величина, характеризующая дальность препятствия от робота. В широко известном методе динамического окна (dynamic window approach [77]) препятствия описываются как отрезки прямых, которые строятся на базе сенсорных данных ультразвуковых датчиков.

В подавляющем большинстве рассмотренных алгоритмов робот представляется как движущаяся окружность, реже – как многоугольник [70]. Динамические препятствия также представляются как окружности. Это обусловлено тем, что окружности позволяют наиболее просто определять ближайшее расстояние от робота до препятствия, что снижает требования к вычислительным ресурсам робота. Если используется сетчатая карта, то препятствия отображаются как множества занятых ячеек карты [13; 68].

Обзор показал, что для объезда подвижных препятствий локальный планировщик робота должен знать *координаты препятствий*, *векторы их скоростей* (проекции скоростей на оси системы координат) и *радиусы* габаритных окружностей. Обычно этого достаточно, чтобы избежать столкновения и добраться до целевой позиции. В некоторых алгоритмах требуется набор точек траектории движения препятствия [97]. Однако существуют способы прогнозирования траектории препятствия по его текущей скорости, положению и кинематической модели.

В современных алгоритмах объезда препятствий учитываются кинематические особенности колёсных мобильных роботов, поскольку это может

сужать область допустимых движений робота. Также большинство методов учитывают динамические характеристики роботов, в частности, разгон и торможение. Динамические характеристики МР требуют учёта допустимых областей движения, что снижает вероятность столкновения робота с препятствием.

Итак, для решения задачи объезда препятствий в частично известной среде предлагается следующее разделение вычислительной нагрузки между ТМ и СМБД. СМБД должен выполнять следующие основные функции:

1. Сканирование пространства на предмет обнаружения препятствий и исключение «ложных» точек скана, т.е. тех точек, которые не относятся к препятствиям (фильтрация точек). Поле зрения сенсорной системы должно быть не менее 180° и покрывать зону робота в направлении его движения.
2. Решение задачи SLAM.
3. Разделение всех препятствий на две группы (рисунок 2.5): статические и динамические (подвижные). Каждое динамическое препятствие должно описываться движущейся габаритной окружностью. Каждое статическое препятствие может описываться набором отрезков прямых или точек.
4. Расчёт скорости каждого подвижного препятствия.

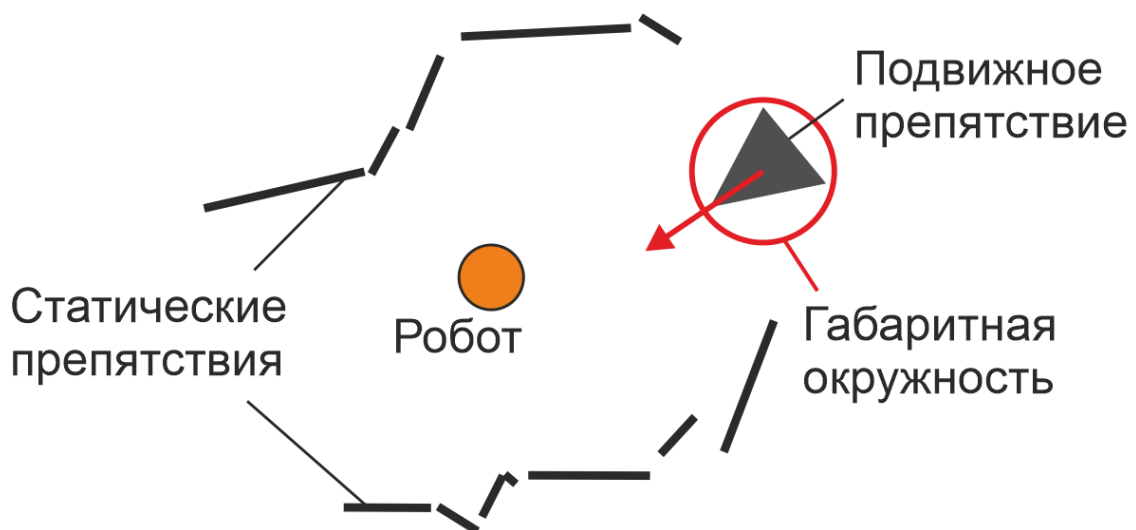


Рисунок 2.5. Модели статических и динамических препятствий

Решение задачи SLAM СМБД должен выполнять совместно с МИУ, который хранит глобальную карту местности, а также при использовании данных, получаемых от сенсорного модуля дальнего действия (СМДД).

ТМ не может выполнять задачу локализации, поскольку его датчики не обладают необходимой точностью. Обнаружение и идентификацию подвижных препятствий может выполнить только отдельный сенсорный модуль (СМБД или СМДД), поскольку он оснащён соответствующими датчиками и выполняет локализацию робота, без которой невозможно отличить подвижное препятствие от неподвижного, если робот перемещается.

ТМ в свою очередь должен выполнять следующие функции:

1. Локальное планирование в пределах ближней зоны СМБД.
2. Объезд препятствий.
3. Управление приводами движителей.
4. Опрос низкоуровневых датчиков (например, концевые выключатели для отслеживания факта столкновения или дальномёры для исключения столкновений).

Ближняя зона СМБД может иметь различный размер и форму в зависимости от поставленной задачи, размеров робота и рабочей среды. В данной работе принимается, что ближняя зона ограничена окружностью радиусом 3 м с центром в характеристической точке робота (например, в центре масс робота). При планировании движения робота принимается, что робот описывается габаритной окружностью.

Алгоритм объезда препятствий теоретически может быть любым, если информация, получаемая от СМБД *достаточна для выполнения алгоритма*. Следует учитывать, что время на объезд препятствия ограничено сверху предельным временем прибытия к цели ($t_{\max}$).

В рамках межмодульного взаимодействия сенсорный модуль ближнего действия отправляет в транспортный модуль следующую структуру данных (таблица 2.4).

Таблица 2.4. Структура данных, отправляемая СМБД в ТМ

Поле данных	Тип данных
Координата X_r текущей позиции робота относительно базовой системы координат, м	Число с плавающей запятой
Координата Y_r текущей позиции робота относительно базовой системы координат, м	Число с плавающей запятой
Набор отрезков прямых, описывающих статические препятствия: $\{I_1, I_2, \dots, I_n\}$, где $I_i = (x_{i1}, y_{i1}, x_{i2}, y_{i2})$ – параметры i -го отрезка прямой: x_{i1}, y_{i1} – координаты первого конца отрезка, x_{i2}, y_{i2} – координаты второго конца отрезка; n – количество отрезков прямых.	Двумерный массив чисел с плавающей запятой или массив кортежей
Набор данных о подвижных препятствиях: $\{O_1, O_2, \dots, O_m\}$, где $O_i = (x_{Oi}, y_{Oi}, v_{xi}, v_{yi}, r_{Oi})$ – упорядоченный набор параметров i -го препятствия, m – количество препятствий в ближней зоне робота, x_{Oi}, y_{Oi} – координаты центра габаритной окружности препятствия с радиусом r_{Oi} в базовой системе координат, v_{xi}, v_{yi} – проекции вектора скорости препятствия на оси базовой системы координат.	Двумерный массив чисел с плавающей запятой или массив кортежей

Представление статических препятствий в виде отрезков прямых является достаточно распространённым подходом в мобильной робототехнике. Использование отрезков в большинстве случаев позволяет с приемлемой точностью описывать препятствия в городской среде и помещениях. При этом статические препятствия можно представлять в виде облака точек [56], что удобно с точки зрения использования практически любого метода объезда препятствий. Однако с ростом разрешающей способности сенсорной системы СМБД возрастает количество точек сканированных препятствий.

Количество отрезков прямых, которые СМБД формирует из облака точек, полученного после сканирования пространства, зависит от сложности самой среды, разрешающей способности сенсорной системы СМБД и алгоритма аппроксимации (метода извлечения прямых из точек). В данной работе не рассматривается конкретный алгоритм аппроксимации, который должен использоваться в СМБД; при этом рекомендуется выбирать алгоритм, обеспечивающий высокую скорость извлечения отрезков прямых и точность, например метод *Split-and-Merge* [99].

Каждый отрезок прямой описывается четырьмя параметрами: координатами концов отрезка, заданными относительно системы координат ТМ. При этом согласно стандарту IEEE 1873-2015 [84], в котором определены модели и форматы представления двумерных карт местности в робототехнике, отрезок прямой описывается параметрами ρ , α , ψ_a и ψ_b (рисунок 2.6). Оба варианта эквивалентны, однако в настоящей работе используются координаты концов отрезков, что обусловлено удобством их применения в алгоритме объезда препятствий, который используется в СУ ТМ (рассматривается в следующей главе).

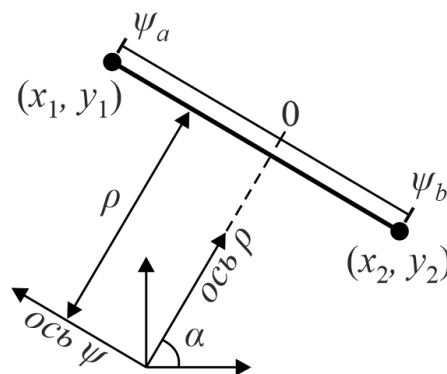


Рисунок 2.6. Параметры для описания отрезка прямой препятствия

Для определения требуемой пропускной способности информационного канала ТМ $\Leftrightarrow$ СМБД необходимо оценить количество отрезков прямых n , информация о которых может содержаться в одном сообщении СМБД. В статье [48] авторов Nguyen V., Gächter S. и др. приводится сравнение характеристик наиболее известных алгоритмов извлечения отрезков из сканов,

получаемых с помощью лазерных лидаров. Сравнение алгоритмов проводилось с использованием данных реальных сканов, полученных с помощью лидара SICK LMS291-S05, который был установлен на мобильном роботе. В статье показано, что количество отрезков, извлечённых алгоритмами, работающими в реальном времени, не превышает 725. Такое число было получено после обработки 100 сканов большого помещения с размером карты 80 на 50 метров, в котором находились различные объекты: стулья, столы, стены, шкафы, окна. Также рядом с роботом оказывались люди. Карта этого помещения (лаборатории) и приблизительная траектория робота представлены на рисунке 2.7.



Рисунок 2.7. Карта помещения, в котором авторы статьи [48] проводили эксперимент

В данном случае СМБД должен отправлять информацию о препятствиях, которые находятся в пределах *ближней* зоны с радиусом 3 м. Для зоны с такими размерами можно предположить, что максимальное число извлечённых отрезков прямых не будет превышать 100-150 отрезков. Такая оценка основана на том, что ближняя зона СМБД охватывает лишь небольшую часть среды мобильного робота.

Прогнозирование числа подвижных препятствий, которые могут находиться вблизи робота, является нетривиальной задачей и в диссертации не рассматривается. В данной работе принимаем, что модульный МР находится в

такой среде, где максимальное количество подвижных препятствий в ближней зоне не превышает 10.

Частота, с которой СМБД должен отправлять данные в ТМ, в первую очередь зависит от быстродействия модульного робота (в частности, от скорости перемещения ТМ), характеристик дальномеров СМБД и времени обработки данных. Для оценки требуемой частоты отправки сенсорных данных будем исходить из следующей ситуации. Пусть модульный МР движется прямолинейно навстречу статическому препятствию со скоростью $v_r \leq 2$ м/с (рисунок 2.8). СМБД должен отправлять сенсорные данные в ТМ с такой частотой, чтобы ТМ (робот) успел остановиться перед препятствием на безопасном расстоянии s . В данном случае принимаем, что $s = 10$ см. Тогда при заданной скорости робота частота передачи сенсорных данных (таблица 2.4) f_{sensor} должна быть *не меньше* 20 Гц. Это значение получено при допущении, что СМБД мгновенно получает новые данные о препятствии, а ТМ мгновенно обрабатывает сенсорные данные и реагирует на них. Соответственно при большей скорости робота и меньшем безопасном расстоянии s частота передачи f_{sensor} должна быть ещё выше. При этом для лазерных дальномеров типовая частота сканирования составляет 10-30 Гц [112], хотя может достигать и 150 Гц, что удовлетворяет приведённому требованию.

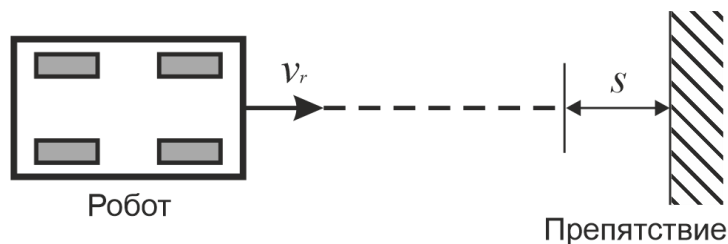


Рисунок 2.8. Иллюстрация движения МР к препятствию, перед которым МР должен остановиться на безопасном расстоянии s

Транспортный модуль должен отправлять в СМБД информацию о своём положении и скорости. Это необходимо для повышения точности работы алгоритмов SLAM. ТМ рассчитывает своё положение и скорость с помощью одометрии. Частота отправки данных должна быть не меньше частоты f_{sensor} .

Структура отправляемых данных имеет вид, указанный в таблице 2.5.

Таблица 2.5. Структура данных, отправляемая ТМ в СМБД

Поле данных	Тип данных
Координата X_r текущей позиции относительно базовой системы координат, м	Число с плавающей запятой
Координата Y_r текущей позиции относительно базовой системы координат, м	Число с плавающей запятой
Скорость ТМ в направлении оси x системы координат ТМ, м/с	Число с плавающей запятой
Скорость ТМ в направлении оси y системы координат ТМ, м/с	Число с плавающей запятой

Отправка информации о скорости транспортным модулем в СМБД может быть полезна для регулирования настроек датчиков сенсорного модуля. Например, если скорость ТМ ниже заданного порога, то следует снизить частоту опроса дальномеров для уменьшения трафика межмодульного взаимодействия и энергопотребления СМБД. При низкой скорости работа СМБД может не отправлять данные о подвижных препятствиях, которые находятся дальше некоторого заданного расстояния.

2.5 Выводы по второй главе

Разработанная функционально-модульная двухуровневая архитектура системы управления мобильного робота обеспечивает реализацию распределённых вычислений, что позволяет снизить вычислительную нагрузку на вычислительное устройство каждого модуля, являющегося самостоятельным устройством со своей системой управления и программно-аппаратными аспектами.

Для реализации функционально-модульного разделения структуры робота на модули был предложен принцип полной функциональности, который позволяет существенно снизить трафик межмодульного информационного

взаимодействия, а также использовать в работе модули, которые имеют различную аппаратно-программную реализацию, но выполняют одну функцию (например, транспортную) и обладают соответствующими программными интерфейсами.

Анализ информационного взаимодействия основных полнофункциональных модулей позволил определить структуры пересылаемых сообщений и их формат между МИУ и ТМ, а также между ТМ и СМБД. Показано, как задача навигации колёсного модульного МР в среде с подвижными и неподвижными препятствиями должна распределяться между МИУ, ТМ и СМБД.

Показано, что структуры данных, пересылаемых между МИУ и ТМ, имеют достаточно простую форму: шесть числовых полей для структуры-задания и три числовых поля с флагом статуса для структуры-отчёта. Такая простота обусловлена использованием предложенного принципа полной функциональности, что позволяет снизить межмодульный трафик интерфейса ТМ $\Leftrightarrow$ МИУ.

На основе анализа типовых алгоритмов объезда подвижных и неподвижных препятствий сформулированы требования к функционалу ТМ и СМБД и их межмодульному интерфейсу ТМ $\Leftrightarrow$ СМБД.

СМБД должен отправлять в ТМ координаты робота, набор параметров отрезков прямых, описывающих статические препятствия и набор данных о подвижных препятствиях с частотой не ниже 20 Гц. Каждое подвижное препятствие должно представляться в виде габаритной окружности. Набор данных о подвижном препятствии должен включать в себя координаты центра окружности, проекции вектора скорости препятствия и радиус окружности. Межмодульный интерфейс ТМ $\Leftrightarrow$ СМБД должен обеспечивать передачу сообщений СМБД, которые могут содержать данные о 10 подвижных препятствиях и 150 отрезках прямых, относящихся к статическим препятствиям.

ГЛАВА 3. СИСТЕМА УПРАВЛЕНИЯ ТРАНСПОРТНОГО МОДУЛЯ

В главе рассматривается предложенная автором формальная модель взаимодействия полнофункциональных модулей с применением теории конечных автоматов [56] и диаграмм состояний унифицированного языка UML. Теория конечных автоматов позволяет исследовать модули как аппаратно-программные агенты, меняющие своё состояние и поведение в зависимости от внешних и внутренних событий [56].

Рассмотрено управление движением транспортного модуля в среде с подвижными и неподвижными препятствиями. Основная задача ТМ – достичь целевой позиции, задаваемой МИУ. Для этого могут быть использованы различные методы, например, пропорциональный регулятор скорости или предложенный автором метод с использованием потенциальных полей [3], задаваемых двумерной функцией Гаусса.

Для объезда динамических препятствий решено использовать классический метод «скоростных препятствий» (Velocity Obstacles [76]), частично дополненный для учёта статических препятствий. Данный метод лучше приспособлен для голономных мобильных роботов, поэтому транспортный модуль рассматривается как движущаяся платформа с четырьмя механум-колёсами.

Были определены требования к программным межмодульным интерфейсам ТМ исходя из объёма пересылаемых данных в процессе взаимодействия ТМ⇔МИУ и ТМ⇔СМБД. Даны рекомендации по выбору коммуникационных каналов и топологии сети полнофункциональных модулей.

Учитывая, что приходящаяся на ТМ вычислительная нагрузка всё ещё велика, было предложено рассматривать ТМ также в виде модульной архитектуры, состоящей из узлов-субмодулей [58]. В такой структуре задачи управления исполнительными приводами и механизмами распределены между субмодулями – реализуются распределённые вычисления. При этом появляется

возможность реконфигурации на уровне транспортной платформы за счёт увеличения числа субмодулей или изменения их состава.

3.1 Модель взаимодействия полнофункциональных модулей

Для разработки системы управления ТМ сначала необходимо составить формальную модель, описывающую взаимодействие ТМ, МИУ и СМБД (см. рисунок 2.1). Основная задача ТМ – достичь целевой позиции, задаваемой МИУ. Модель взаимодействия позволит определить требования к программному обеспечению ТМ и упростит разработку компьютерной модели.

Математическая модель мобильного робота с модульной архитектурой должна учитывать взаимодействие модулей и их функционирование как полнофункциональных интеллектуальных агентов. В данной работе предлагается использовать теорию конечных автоматов для формального описания основных частей-модулей системы управления мобильного робота. Основное внимание уделяется транспортному модулю и модулю интеллектуального управления.

Конечный автомат (КА) является удобным математическим инструментом для описания систем с конечным множеством состояний. КА активно используются при разработке программного обеспечения (например, «лексические анализаторы» компиляторов, управляющие программы микроконтроллеров) и цифровых электрических схем современных вычислительных машин.

Конечный автомат позволяет описать поведение некоторой технической системы, которая в каждый дискретный момент времени находится только в одном состоянии [11]. КА принимает различные входные сигналы и формирует выходные. Состояние автомата в следующий момент времени зависит от входного сигнала и предыдущего состояния. Выходной сигнал может являться функцией входного сигнала и/или состояния. Если выходной сигнал зависит

только от состояния, то КА является *автоматом Мура*; если выходной сигнал зависит и от состояния, и от входа, то такой КА считается *автоматом Мили*.

В настоящей работе рассматриваются только *детерминированные* КА, т.е. такие КА, в которых для каждой комбинации состояния и входного сигнала есть только одно уникальное следующее состояние.

КА удобно описывать графически: с помощью таблицы или диаграммы состояний. В таблице новое состояние и соответствующий выход ставятся в соответствие предыдущему состоянию и полученному входу (рисунок 3.1).

Вход \ Состояние	a	b	c
$S1$	$S2/x$	$S2/y$	$S1/x$
$S2$	$S1/x$	$S3/z$	$S2/z$
$S3$	...	...	...

x, y, z - выходные сигналы

Рисунок 3.1. Пример типовой таблицы конечного автомата с входами a, b, c и выходами x, y, z

Недостатком такой таблицы является сложность представления выходных сигналов (*действий* системы) при различных условиях, например, когда необходимо описать действия автомата *при входе* в некоторое состояние, или *во время пребывания* в состоянии.

Диаграмма состояний автомата представляет собой размеченный ориентированный граф (рисунок 3.2).

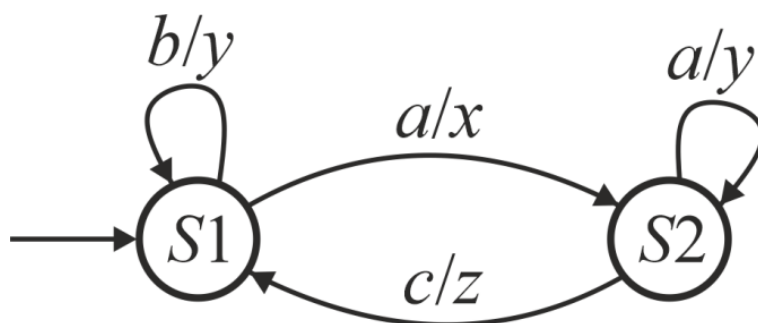


Рисунок 3.2. Пример типовой диаграммы состояний КА

Вершины графа – состояния КА, дуги – переходы, метки над дугами вида a/x обозначают входной сигнал a , вызвавший переход и выходной сигнал x (если это автомат Мили или Мура).

Классическая диаграмма состояний не всегда удобна для описания комплексных технических систем, применяемых на практике, поскольку добавление новых функций автомата быстро ведёт к взрывному росту числа его состояний. Кроме того, диаграмма не позволяет учитывать такие свойства систем, как иерархичность, модульность и параллельность [80].

В диссертации предлагается использовать унифицированный язык UML [47] для описания диаграмм состояний конечных автоматов (UML state machine). Диаграммы состояний UML широко используются для разработки спецификаций программ, коммуникационных протоколов, моделирования поведения технических систем и др. Конечные автоматы UML сочетают в себе свойства автоматов Мура и Мили. Диаграммы UML позволяют точнее описывать поведение КА и его действия (выходные сигналы) в зависимости от различных событий по сравнению с классическими диаграммами.

События могут быть как внешними по отношению к автомату, так и внутренними. *Внешние события* порождаются внешним по отношению к автомату объектом: пришло сообщение или команда от другого модуля, поступил сигнал об изменении состояния объекта. *Внутренние события* порождаются самим автоматом: завершилось выполнение какого-либо действия, произошёл отказ узла системы, истекло время нахождения в некотором состоянии – временное событие (time event) в терминах UML.

Рассмотрим конечные автоматы транспортного модуля и модуля интеллектуального управления.

Транспортный модуль робота может находиться в двух обобщённых состояниях (рисунок 3.3):

- включён ($S_{T.1}$),
- выключен ($S_{T.0}$).

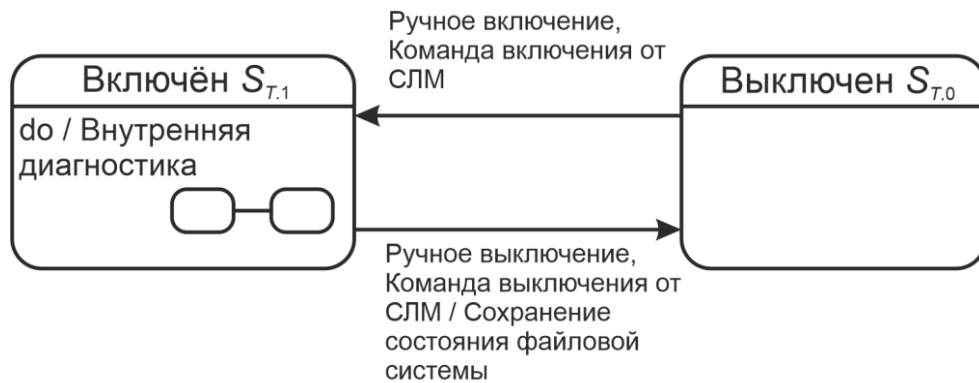


Рисунок 3.3. Обобщённые состояния ТМ

На диаграмме показано, что после ручного выключения или выключения по команде от силового модуля транспортный модуль должен выполнить сохранение файловой системы (корректное выключение всех вычислительных устройств). В общем случае текст над дугой-переходом имеет следующую форму:

<список событий> [сторожевое условие] / <список действий>.

Переход выполняется, когда произошло одно или несколько событий из списка, и истинно сторожевое условие.

Спецификатор поведения «do» внутри состояния обозначает действия (после символа «/»), которые автомат выполняет постоянно, находясь в этом состоянии. В данном случае ТМ в фоновом режиме должен проводить диагностику своих основных узлов-субмодулей (рассматриваются в конце этой главы).

В диаграммах UML часто используется спецификатор «entry», который обозначает действия, выполняемые автоматом при входе в состояние.

Во *включенном состоянии* можно выделить следующие *подсостояния* (substates):

- рабочий режим ($S_{T.1.1}$),
- аварийный режим ($S_{T.1.2}$),
- неисправность ($S_{T.1.3}$).

Тогда в терминах UML состояние «Включён» является *составным состоянием* (composite state), поскольку состоит из нескольких *подсостояний*.

Состояние «рабочий режим» также является составным, поскольку включает в себя три подсостояния:

- ожидание цели ($S_{T.1.1.0}$),
- перемещение к цели ($S_{T.1.1.1}$),
- цель достигнута ($S_{T.1.1.2}$).

Рассмотрим диаграмму UML автомата ТМ (рисунок 3.4) для составного состояния «Включён» ($S_{T.1}$) (полная диаграмма не приводится из-за её габаритов).

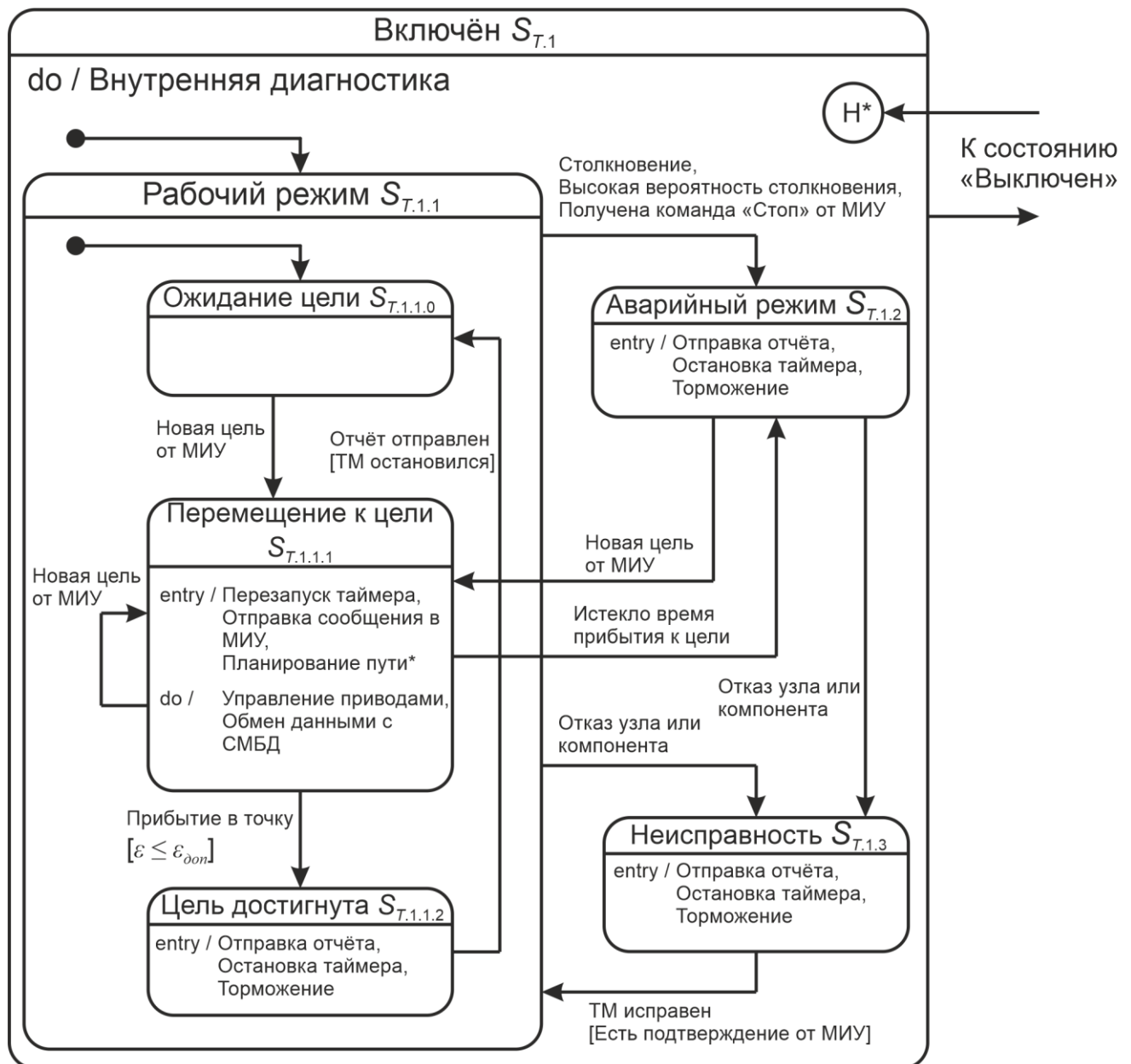


Рисунок 3.4. Диаграмма состояний UML для составного состояния «Включён» транспортного модуля

После включения ТМ сразу переходит в подсостояние «Рабочий режим» ($S_{T.1.1}$). Из этого состояния автомат может перейти в «Аварийный режим» ($S_{T.1.2}$), если произошло столкновение, существует высокая вероятность столкновения или получена команда «Стоп» от МИУ (например, другой модуль робота неисправен и нужно срочно прекратить движение). Переход в $S_{T.1.2}$ также возможен из подсостояния $S_{T.1.1.1}$, если истекло время прибытия к цели. Если в процессе работы обнаружился отказ узла или компонента ТМ, автомат должен перейти в состояние «Неисправность» ($S_{T.1.3}$).

При входе в состояние «Аварийный режим» ТМ должен отправить отчёт в МИУ, остановить таймер для отсчёта времени, затрачиваемого на перемещение и выполнить торможение. Если фиксируется факт столкновения, ТМ может выполнить алгоритм отъезда в безопасном направлении (на диаграмме не отмечено, поскольку это зависит от конкретной реализации и ситуации). Чтобы вернуться в состояние «Рабочий режим» (в подсостояние $S_{T.1.1.1}$), ТМ должен получить новую цель от МИУ.

При входе в состояние «Неисправность» ТМ выполняет те же действия, что и при входе в «Аварийный режим». Для возвращения в рабочее состояние ТМ должен быть полностью исправен, а также необходимо подтверждение от МИУ о продолжении работы (*сторожевое условие* на соответствующем переходе).

На диаграмме присутствует специальный символ H^* с окружностью, который представляет собой *историческое псевдосостояние с глубокой историей* (deep history pseudostate). Оно используется для сохранения конфигурации автомата в момент выхода из состояния «Включён» после получения команды на отключение ТМ. Например, пусть в некоторый момент времени ТМ находился в состоянии $S_{T.1.1.1}$ («Перемещение к цели»). Затем поступила команда на выключение (переход из $S_{T.1}$ в $S_{T.0}$), и историческое псевдосостояние запомнило состояние $S_{T.1.1.1}$. Тогда после повторного включения ТМ (переход из $S_{T.0}$ в $S_{T.1}$), автомат ТМ сразу окажется в состоянии $S_{T.1.1.1}$.

При входе в «Рабочий режим» автомат переходит в исходное подсостояние «Ожидание цели» ($S_{T.1.1.0}$). Из этого подсостояния возможен переход в

подсостояние «Перемещение к цели», если получена новая точка от МИУ (внешнее событие).

Состояние «Перемещение к цели» ($S_{T.1.1.1}$) отражает процесс перемещения ТМ к очередной точке, полученной от МИУ. При входе в это состояние система управления ТМ должна перезапустить таймер для отсчёта времени, затрачиваемого на перемещение к цели, отправить сообщение в МИУ о том, что начато движение, и, если необходимо, спланировать маршрут. Необходимость планирования пути определяется алгоритмом навигации ТМ.

Находясь в состоянии перемещения к цели, ТМ выдаёт задающие воздействия на приводы (субмодули) и обменивается данными с СМБД (см. главу 2). Переход в то же самое состояние $S_{T.1.1.1}$ возможен, когда необходимо немедленно начать перемещение по другому пути в соответствии с планом МИУ.

Автомат переходит в состояние «Аварийный режим»², если истекло время прибытия к цели:

$$t > t_{\max},$$

где t – текущее время с момента начала перемещения,

$t_{\max}$ – дедлайн (предельное время прибытия к цели); задаётся модулем интеллектуального управления.

Переход в состояние «Цель достигнута» ($S_{T.1.1.2}$) происходит, когда по оценке СУ ТМ прибытие в точку выполнено с допустимой погрешностью (сторожевое условие перехода):

$$\varepsilon \leq \varepsilon_{\text{дон}},$$

где $\varepsilon_{\text{дон}}$ – величина допустимой ошибки по положению в целевой точке,

ε – Евклидово расстояние от текущего положения ТМ с координатами X_r, Y_r до целевой точки с координатами X_i, Y_i .

При входе в состояние $S_{T.1.1.2}$ автомат ТМ должен отправить отчёт в МИУ, остановить таймер и выполнить торможение. После успешной отправки отчёта и останова (сторожевое условие), ТМ вновь переходит в исходное состояние $S_{T.1.1.0}$.

² Выход из подсостояния означает одновременный выход из составного состояния по правилам UML

На диаграмме 3.4 видно, что автомат ТМ получает от МИУ команды, а в ответ отправляет своё состояние (отчёт) и статус. Это соответствует принципу полной функциональности и общим рекомендациям по разработке систем из КА, взаимодействующих в режиме «ведущий-ведомый» (Master-Slave) [92].

Рассмотрим конечный автомат МИУ. Как и ТМ модуль интеллектуального управления может находиться в двух обобщённых состояниях: включён ($S_{M.1}$) и выключен ($S_{M.0}$). Диаграмма UML для этих состояний аналогична рисунку 3.3.

Состояние «Включён» содержит следующие подсостояния:

- рабочий режим ($S_{M.1.1}$),
- аварийный останов ($S_{M.1.2}$),

Составное состояние $S_{M.1.1}$ в свою очередь включает в себя подсостояния:

- ожидание задачи ($S_{M.1.1.0}$)
- планирование пути ($S_{M.1.1.1}$),
- контроль ($S_{M.1.1.2}$),
- задача выполнена ($S_{M.1.1.3}$).

Рассмотрим диаграмму UML автомата МИУ (рисунок 3.5) для составного состояния «Включён» ($S_{M.1}$) (полная диаграмма не приводится из-за её габаритов). После включения МИУ (всего робота) автомат сразу переходит в подсостояние «Рабочий режим» ($S_{M.1.1}$). Из этого состояния возможен переход в состояние «Аварийный останов» ($S_{M.1.2}$), если произошло одно из следующих событий:

- один из модулей неисправен,
- силовой модуль (СЛМ) сообщает о низком уровне заряда.

Также возможен переход из подсостояния $S_{M.1.1.1}$ (см. ниже).

При входе в состояние $S_{M.1.2}$ МИУ должен отправить отчёт своему супервизору (например, человеку-оператору), содержащий информацию о состоянии модулей робота и текущем статусе выполнения задания. Также при входе в состояние МИУ должен отправить команду на приостановление работы всех модулей (кроме силового модуля).

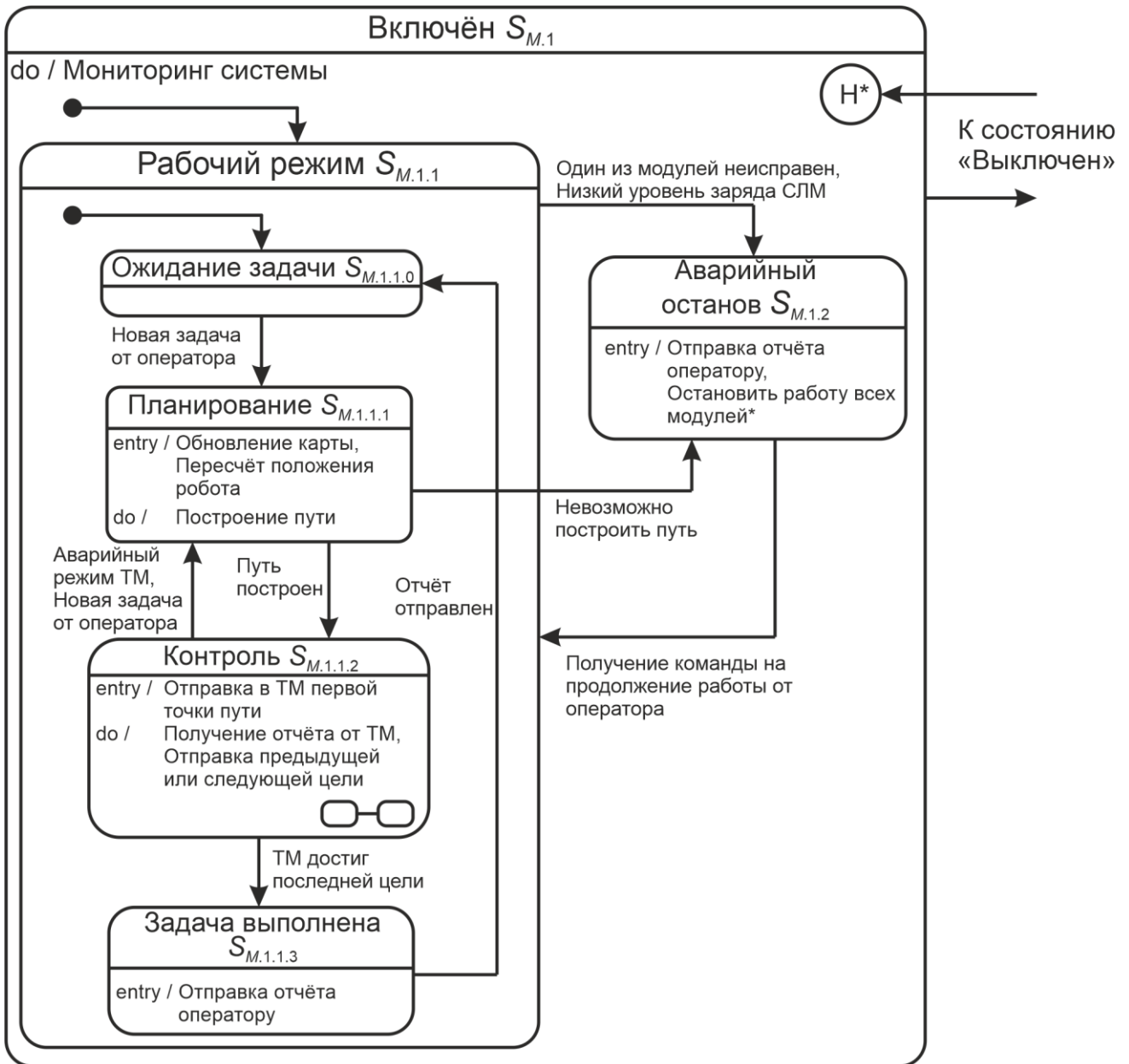


Рисунок 3.5. Диаграмма состояний UML для составного состояния «Включён» модуля интеллектуального управления

Аналогично автомату ТМ диаграмма состояний МИУ содержит историческое псевдосостояние N^* для сохранения конфигурации модуля в момент выключения.

Рассмотрим составное состояние $S_{M.1.1}$.

В исходном подсостоянии «Ожидание задачи» ($S_{M.1.1.0}$) МИУ находится до тех пор, пока не получит новую задачу от оператора.

Состояние «Планирование» ($S_{M.1.1.1}$) отражает процесс построения пути на карте местности. В момент входа в это состояние МИУ должен обновить карту

(используя данные от СМБД и СМДД) и локализовать робот. Находясь в этом состоянии, МИУ планирует путь к цели, заданной оператором. Как только путь построен, выполняется переход в состояние «Контроль» ($S_{M.1.1.2}$). Существует вероятность, что путь построить невозможно, тогда автомат должен выполнить «Аварийный останов».

Состояние «Контроль» ($S_{M.1.1.2}$) в общем случае может являться составным, поскольку логика анализа отчётов ТМ зависит от конкретной реализации модуля интеллектуального управления. При входе в состояние МИУ должен отправить в ТМ данные первой точки маршрута (цель). Далее находясь в этом состоянии, МИУ периодически получает отчёты ТМ и анализирует их. Если очередная цель достигнута с допустимой ошибкой ε_{don} , МИУ отправляет следующую цель, иначе отправляет предыдущую. В этом состоянии должен быть реализован счётчик ошибок ТМ. Превышение заданного порога счётчика ошибок говорит о том, что ТМ неисправен, при условии, что исправен сенсорный модуль ближнего действия. Если ТМ достиг последней цели, то выполняется переход в состояние «Задача выполнена» ($S_{M.1.1.3}$). При получении отчёта об аварийном режиме ТМ, МИУ должен заново спланировать путь (переход в $S_{M.1.1.1}$).

При входе в состояние $S_{M.1.1.3}$ автомат МИУ должен отправить отчёт оператору, и после успешной отправки перейти в исходное состояние $S_{M.1.1.0}$.

Следует отметить, что данная диаграмма состояний составлена при условии, что модульный робот состоит только из четырёх модулей: МИУ, ТМ, СМБД и СЛМ. При этом состояние $S_{M.1.1}$ значительно усложнится, если в состав модульного робота будет входить, например, модуль манипуляторов. В этом случае $S_{M.1.1}$ будет включать в себя два *параллельных* составных подсостояния, в каждом из которых будут свои процессы планирования и контроля.

Компьютерные модели автоматов транспортного модуля и модуля интеллектуального управления, их реализация и результаты компьютерных симуляций рассматриваются в главе 5.

3.2 Динамическая модель транспортного модуля

В диссертации в качестве примера наземного РТК рассматривается колёсный транспортный модуль с всенаправленным мобильным механизмом. ТМ представляет собой движущуюся мобильную платформу с четырьмя меканум-колёсами. Меканум-колесо – это всенаправленное колесо, у которого ролики, установленные на наружной поверхности, ориентированы под углом 45 градусов к основной оси колеса.

Такая мобильная платформа является голономной, её основное преимущество по сравнению с типовыми неголономными платформами заключается в возможности перемещения в любом направлении без изменения угловой ориентации платформы. При этом многие алгоритмы объезда препятствий лучше работают с голономными платформами. Основной недостаток – по сравнению с обычными колёсами меканум-колёса сильнее проскальзывают, что значительно ухудшает расчёт положения робота с помощью одометрии. Однако в данной модульной системе положение мобильного робота определяется в основном за счёт датчиков СМБД.

Схема ТМ с всенаправленным мобильным механизмом показана на рисунке 3.6.

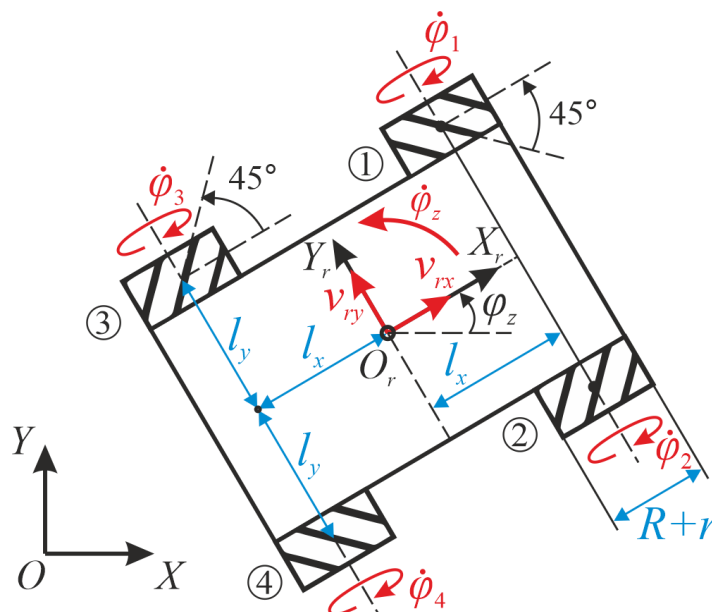


Рисунок 3.6. Схема ТМ с всенаправленным механизмом

Обозначения, принятые на схеме: OXY – базовая система координат, $O_r X_r Y_r$ – система координат, связанная с мобильной платформой ТМ, $\dot{\phi}_1, \dot{\phi}_2, \dot{\phi}_3, \dot{\phi}_4$ – угловые скорости колёс, v_{rx}, v_{ry} – проекции линейной скорости ТМ на оси системы координат, связанной с мобильной платформой, ϕ_z – угол ориентации мобильной платформы относительно базовой системы координат, R – радиус меканум-колеса, r – радиус ролика меканум-колеса, l_x – расстояние от характеристической точки платформы O_r совпадающей с центром масс до каждой из осей передних и задних колёс, расстояние между центрами колёс составляет $2l_y$.

Кинематическая модель такой мобильной платформы описывается следующим образом [83]. Зависимость скоростей колёс $\dot{\phi}_1, \dot{\phi}_2, \dot{\phi}_3, \dot{\phi}_4$ от скоростей платформы определяется выражением:

$$\dot{\phi} = \mathbf{J} \mathbf{v}_r, \quad (3.1)$$

где $\dot{\phi}$ – вектор угловых скоростей колёс,

$\mathbf{J}$ – матрица 4 на 3,

$\mathbf{v}_r$ – вектор скорости мобильной платформы:

$$\dot{\phi} = \begin{pmatrix} \dot{\phi}_1 \\ \dot{\phi}_2 \\ \dot{\phi}_3 \\ \dot{\phi}_4 \end{pmatrix}, \quad \mathbf{J} = \frac{1}{R+r} \begin{pmatrix} 1 & -1 & -(l_x + l_y) \\ 1 & 1 & l_x + l_y \\ 1 & 1 & -(l_x + l_y) \\ 1 & -1 & l_x + l_y \end{pmatrix}, \quad \mathbf{v}_r = \begin{pmatrix} v_{rx} \\ v_{ry} \\ \dot{\phi}_z \end{pmatrix}.$$

Для выражения вектора скорости платформы $\mathbf{v}_r$ (решения прямой задачи о скорости) используется псевдообратная матрица $\mathbf{J}^+ = (\mathbf{J}^T \mathbf{J})^{-1} \mathbf{J}^T$:

$$\mathbf{J}^+ = \frac{R+r}{4} \begin{pmatrix} 1 & 1 & 1 & 1 \\ -1 & 1 & 1 & -1 \\ -1 & 1 & -1 & 1 \\ \frac{1}{l_x + l_y} & \frac{1}{l_x + l_y} & \frac{-1}{l_x + l_y} & \frac{1}{l_x + l_y} \end{pmatrix}.$$

Тогда скорость платформы определяется выражением:

$$\mathbf{v}_r = \mathbf{J}^+ \dot{\phi},$$

соответственно компоненты вектора скорости имеют вид:

$$\begin{aligned}
v_{rx} &= \frac{R+r}{4}(\dot{\phi}_1 + \dot{\phi}_2 + \dot{\phi}_3 + \dot{\phi}_4), \\
v_{ry} &= \frac{R+r}{4}(-\dot{\phi}_1 + \dot{\phi}_2 + \dot{\phi}_3 - \dot{\phi}_4), \\
\dot{\phi}_z &= \frac{R+r}{4(l_x + l_y)}(-\dot{\phi}_1 + \dot{\phi}_2 - \dot{\phi}_3 + \dot{\phi}_4).
\end{aligned}$$

Управление ТМ осуществляется по скорости, поэтому формула (3.1) является ключевой, по которой вычисляются заданные скорости механум-колёс платформы.

Уравнения динамики мобильной платформы с четырьмя механум-колёсами определяются с помощью уравнений Лагранжа второго рода [83]:

$$\mathbf{M} \begin{pmatrix} \ddot{\phi}_1 \\ \ddot{\phi}_2 \\ \ddot{\phi}_3 \\ \ddot{\phi}_4 \end{pmatrix} = \begin{pmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \\ \tau_4 \end{pmatrix} - \begin{pmatrix} N_1 f \operatorname{sgn}(\dot{\phi}_1) \\ N_2 f \operatorname{sgn}(\dot{\phi}_2) \\ N_3 f \operatorname{sgn}(\dot{\phi}_3) \\ N_4 f \operatorname{sgn}(\dot{\phi}_4) \end{pmatrix}, \quad (3.2)$$

где τ_i – крутящий момент электропривода i -го колеса ($i = 1, 2, 3, 4$),

f – коэффициент сопротивления качению,

N_i – сила реакции опоры, действующая на i -е колесо,

$\operatorname{sgn}(x)$ – функция, которая возвращает знак переменной x .

Матрица $\mathbf{M}$ определяется следующим образом:

$$\mathbf{M} = \begin{pmatrix} A+B+C & -B & B & A-B \\ -B & A+B+C & A-B & B \\ B & A-B & A+B+C & -B \\ A-B & B & -B & A+B+C \end{pmatrix}.$$

В матрице $\mathbf{M}$ вспомогательные параметры A, B, C имеют вид:

$$A = \frac{m(R+r)^2}{8}, \quad B = \frac{I_z(R+r)^2}{16(l_x + l_y)^2}, \quad C = I_w,$$

где m – масса всего модульного МР с учётом масс механум-колёс;

I_z – момент инерции всего модульного МР относительно вертикальной оси Z_r системы координат платформы;

I_w – момент инерции механум-колеса относительно собственной оси вращения.

Из формулы (3.2) можно выразить вектор крутящих моментов механум-колёс платформы, что является решением обратной задачи динамики:

$$\begin{pmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \\ \tau_4 \end{pmatrix} = \mathbf{M} \begin{pmatrix} \ddot{\phi}_1 \\ \ddot{\phi}_2 \\ \ddot{\phi}_3 \\ \ddot{\phi}_4 \end{pmatrix} + \begin{pmatrix} N_1 f \operatorname{sgn}(\dot{\phi}_1) \\ N_2 f \operatorname{sgn}(\dot{\phi}_2) \\ N_3 f \operatorname{sgn}(\dot{\phi}_3) \\ N_4 f \operatorname{sgn}(\dot{\phi}_4) \end{pmatrix}.$$

В данных уравнениях динамики не учитываются массы роликов и их моменты инерции. Также принимается, что ТМ перемещается по однородной поверхности, поэтому коэффициент трения качения f одинаков для каждого колеса. Будем считать, что на каждое колесо приходится одинаковая нагрузка, тогда получаем следующее соотношение:

$$N_1 = N_2 = N_3 = N_4 = \frac{G_r}{4},$$

где G_r – вес всего модульного МР.

Рассмотренные уравнения динамики мобильной платформы с механум-колёсами можно выразить в базовой системе координат, что подробно рассматривается в статье [83].

3.3 Движение транспортного модуля к целевой точке

Для перемещения ТМ к заданной цели необходимо выбрать алгоритм движения модуля (состояние $S_{T,1,1,1}$ из диаграммы состояний ТМ). При этом не ставится задача отслеживания какой-либо рассчитанной траектории (хотя это может быть предусмотрено конкретной реализацией СУ ТМ). Важно лишь обеспечить перемещение к i -й точке на плоскости с заданной точностью ε_{don} за отведённое время $t_{\max}$.

Система управления транспортного модуля управляет его движением за счёт регулирования вектора скорости колёсной платформы – вектора

$\mathbf{v}_r = (v_{rx}, v_{ry})^T$, что обеспечивается использованием мобильной платформы с всенаправленным мобильным механизмом как показано в разделе 3.2. При этом не задаётся угловая скорость поворота платформы, поскольку в рассматриваемой задаче угловая ориентация транспортного модуля в точках маршрута не имеет значения (см. раздел 2.2).

Для перемещения ТМ к заданной *фиксированной* точке с координатами X_i, Y_i (рисунок 3.7) можно использовать пропорциональный закон управления скоростью мобильной платформы, который широко распространён в робототехнике [102; 107]:

$${}^b\mathbf{v}_r = \mathbf{K}\mathbf{e} = \begin{pmatrix} k_1 & 0 \\ 0 & k_2 \end{pmatrix} \begin{pmatrix} X_i - X_r \\ Y_i - Y_r \end{pmatrix}, \quad (3.3)$$

где ${}^b\mathbf{v}_r = (\dot{X}_r, \dot{Y}_r)^T$ – скорость ТМ относительно базовой системы координат,

X_r, Y_r – координаты текущей позиции робота (ТМ),

$\mathbf{e}$ – вектор позиционной ошибки $\mathbf{e} = (X_i - X_r, Y_i - Y_r)^T$,

k_1, k_2 – коэффициенты пропорционального регулятора.

В этом случае позиционная ошибка асимптотически стремится к нулю, поскольку справедливо следующее:

$$\dot{\mathbf{e}} = \begin{pmatrix} \dot{X}_i - \dot{X}_r \\ \dot{Y}_i - \dot{Y}_r \end{pmatrix} = - \begin{pmatrix} \dot{X}_r \\ \dot{Y}_r \end{pmatrix} = -\mathbf{K}\mathbf{e}.$$

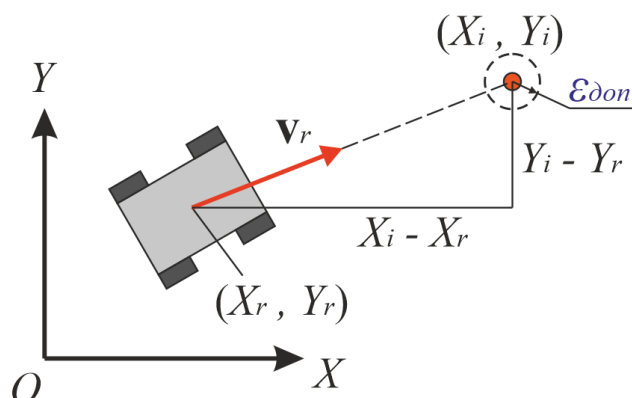


Рисунок 3.7. Схема движения робота к целевой точке

Соответственно вектор скорости $\mathbf{v}_r$ заданный относительно системы координат платформы ТМ задаётся выражением:

$$\mathbf{v}_r = \mathbf{R} \cdot {}^b \mathbf{v}_r = \begin{pmatrix} \cos \varphi_z & \sin \varphi_z \\ -\sin \varphi_z & \cos \varphi_z \end{pmatrix} \begin{pmatrix} k_1 & 0 \\ 0 & k_2 \end{pmatrix} \begin{pmatrix} X_i - X_r \\ Y_i - Y_r \end{pmatrix},$$

где $\mathbf{R}$ – матрица поворота от базовой системы координат OXY к системе координат $O_rX_rY_r$, связанной с ТМ.

Существуют и другие способы движения к заданной точке. Например, в статье [3] автором предложен метод движения к опорным точкам маршрута с помощью потенциальных полей, которые действуют на МР как силы притяжения, отталкивания и вращения. Сила каждого поля задаётся с помощью двумерной функции Гаусса (возможно применение других гладких функций).

В данном случае для i -й целевой точки вычисляется одно потенциальное поле, «притягивающее» к ней ТМ с виртуальной силой F_i (рисунок 3.8):

$$F_i = F_0 \cdot e^{-\left[(w_{xi}(X_{ri}-X_{ci}))^2 + (w_{yi}(Y_{ri}-Y_{ci}))^2 \right]},$$

где F_0 – базовое значение виртуальной силы притяжения,

w_{xi}, w_{yi} – вещественные числа, определяющие ширину потенциального поля в

продольном и поперечном направлениях, при этом ширина поля в

поперечном направлении должна быть меньше;

X_{ri}, Y_{ri} – координаты ТМ (робота) в системе координат потенциального поля

задаваемого двумерной функции Гаусса ($O_i x_i y_i$);

X_{ci}, Y_{ci} – координаты центра функции Гаусса (максимума) в системе координат

$O_i x_i y_i$.

Потенциальное поле, задаваемое функцией Гаусса, повернуто на угол γ_i относительно базовой системы координат.

Линейная скорость рассматриваемого транспортного модуля с всенаправленным мобильным механизмом пропорциональна равнодействующей всех приложенных к ТМ виртуальных сил (в данном случае только одной виртуальной силе F_i).

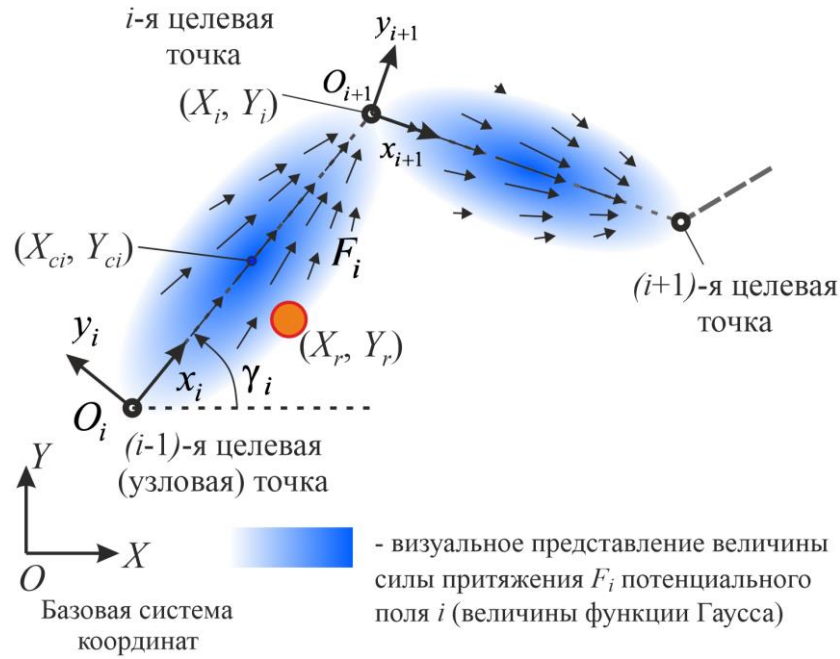


Рисунок 3.8. Визуальное представление потенциальной силы притяжения к i -й целевой точке

Проекции силы F_i на оси системы координат транспортного модуля имеют вид:

$$\begin{aligned} F_{x,i} &= F_i (\cos \alpha_i \cos \varphi_z + \sin \alpha_i \sin \varphi_z), \\ F_{y,i} &= F_i (\sin \alpha_i \cos \varphi_z - \cos \alpha_i \sin \varphi_z), \end{aligned}$$

где α_i – угол направления i -й силы притяжения к i -й целевой точке относительно базовой системы координат.

Окончательно закон управления линейной скоростью ТМ может быть записан следующим образом:

$$\begin{cases} v_{rx} = K_f F_{x,i} \\ v_{ry} = K_f F_{y,i} \end{cases}$$

где K_f – коэффициент пропорциональности потенциальной силы.

Основное преимущество предложенного метода по сравнению с известными методами движения по траектории заключается в обеспечении гладкого и непрерывного закона управления движением ТМ – скорость изменяется плавно и без скачков. При использовании нескольких опорных точек и дополнительных потенциальных полей вращения и отталкивания появляется

возможность двигаться по сложным траекториям *без их предварительного построения* [3].

3.4 Алгоритм объезда подвижных препятствий

Предполагается, что маршрут, построенный МИУ, является безопасным для робота – столкновения со статическими препятствиями исключены. В первую очередь ТМ должен обеспечить объезд *подвижных* препятствий, поскольку их сложнее учесть на карте местности. Однако в процессе объезда при движении к i -й целевой точке транспортный модуль может отклониться от безопасной спланированной траектории, поэтому следует также обеспечить обход статических препятствий. В СУ ТМ должен использоваться такой алгоритм объезда препятствий, для функционирования которого достаточны данные, получаемые от сенсорного модуля ближнего действия (см. таблицу 2.4).

Во второй главе был выполнен анализ различных алгоритмов объезда подвижных препятствий. Несмотря на разнообразие подходов, большинство рассмотренных алгоритмов основано на том, что можно спрогнозировать положение препятствий через некоторый небольшой промежуток времени. Предполагается, что за этот промежуток времени препятствие не успевает изменить вектор скорости. При этом мобильный робот выбирает путь обхода с учётом разгона и торможения.

В данной работе для объезда подвижных препятствий был выбран классический метод «скоростных препятствий» (Velocity Obstacle - VO) [76]. Это один из самых известных алгоритмов, обеспечивающий движение в динамической среде и имеющий большое количество модификаций [64; 70; 98; 110].

Метод формулируется следующим образом. Рассмотрим две движущиеся окружности на плоскости – A (робот) и B (препятствие) (рисунок 3.9). Скорости окружностей – $\mathbf{v}_A$ и $\mathbf{v}_B$ соответственно. Для вычисления VO робот представляется

точкой A , а окружность B увеличивается на радиус окружности робота и обозначается как $\hat{B}$. Тогда радиус окружности $\hat{B}$ равен $r_A + r_B$.

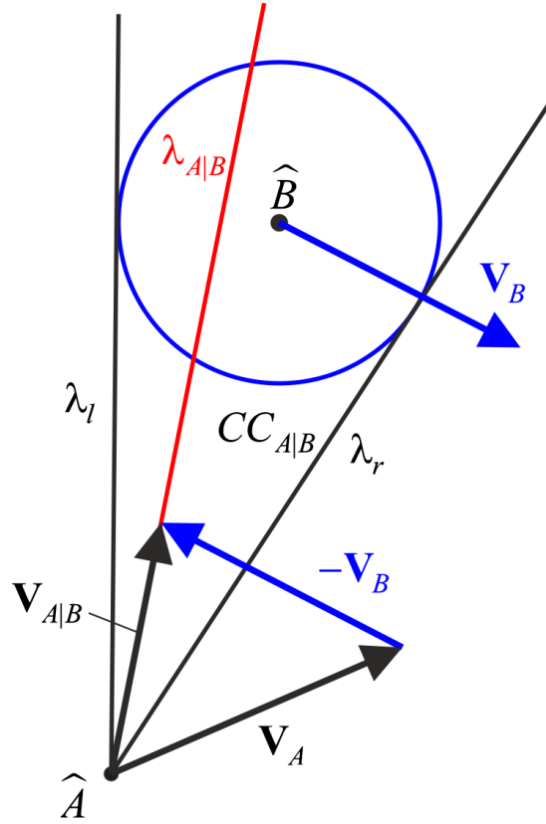


Рисунок 3.9. Конус $CC_{A|B}$ и относительная скорость робота

В основе метода лежит построение конуса относительных скоростей, приводящих к столкновению. Конус представляет собой сектор с вершиной в A , ограниченный двумя касательными λ_l и λ_r к окружности $\hat{B}$ (рисунок 3.9). Любая относительная скорость, лежащая между двумя касательными, приведёт к столкновению через некоторое время, если робот и препятствия будут поддерживать такую скорость. Тогда любая относительная скорость, лежащая за пределами конуса $CC_{A|B}$, гарантирует *отсутствие столкновения*, если препятствие сохранит свою текущую скорость и форму.

Формально конус $CC_{A|B}$ описывается следующим образом:

$$CC_{A|B} = \left\{ \mathbf{v}_{A|B} \mid \lambda_{A|B} \cap B \neq \emptyset \right\},$$

где $\mathbf{v}_{A|B}$ — скорость A относительно скорости B ,

$$\mathbf{v}_{A|B} = \mathbf{v}_A - \mathbf{v}_B,$$

$\lambda_{A|B}$ – прямая вдоль вектора $\mathbf{v}_{A|B}$.

Для учёта нескольких препятствий и удобства реализации метода используются конусы относительных скоростей вместе с абсолютной скоростью робота (относительно базовой системы координат). Для этого конус $CC_{A|B}$ смещается на величину модуля вектора скорости препятствия $\mathbf{v}_B$. В результате получаем «скоростное препятствие» VO (рисунок 3.10):

$$VO_{A|B} = CC_{A|B} \oplus \mathbf{v}_B,$$

где $\oplus$ обозначает сумму Минковского.

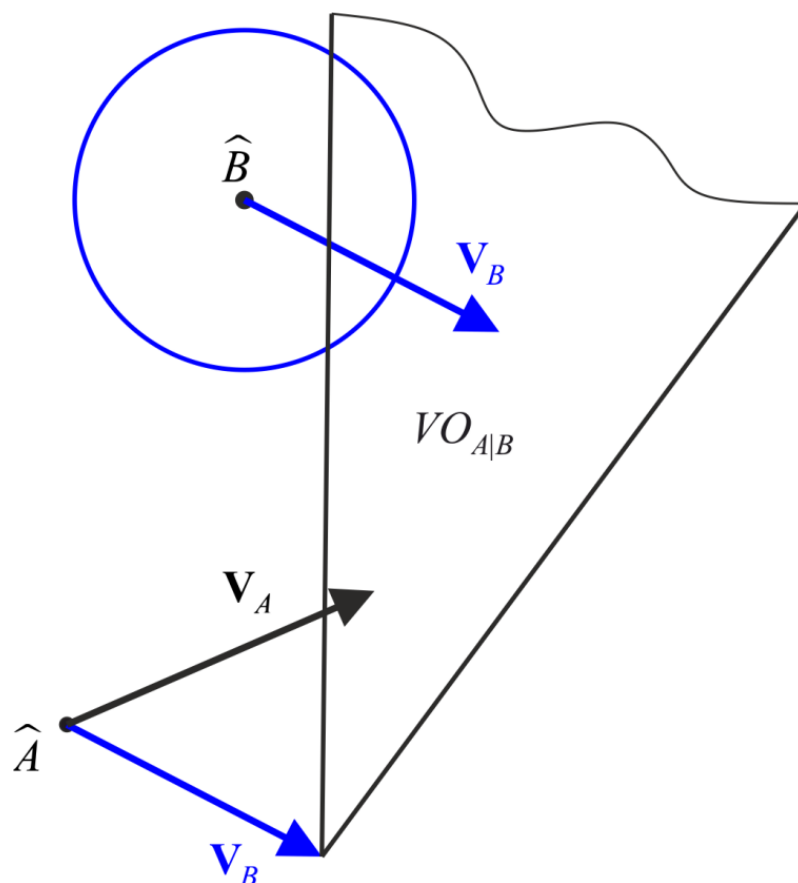


Рисунок 3.10. Скоростное препятствие VO

Для объезда препятствия робот должен выбирать новую скорость за пределами $VO_{A|B}$:

$$A(t) \cap B(t) = \emptyset, \text{ если } \mathbf{v}_A \notin VO_{A|B}(t).$$

Если вектор скорости выбран на границе VO , то робот и препятствие могут касаться друг друга. Для исключения касания следует увеличить размер препятствия на некоторый запас по расстоянию. Для статического препятствия VO совпадает с конусом $CC_{A|B}$.

Для объезда множества препятствий новая скорость робота должна быть за пределами объединения множеств $VO_{A|B_i}$:

$$VO_{A|B_1, B_2, \dots, B_m} = \bigcup_{i=1}^m VO_{A|B_i},$$

где $i = 1, 2, \dots, m$, m – число подвижных препятствий.

В оригинальной статье [76] предлагается учитывать не все препятствия, а только лишь те, которые могут привести к столкновению с роботом в ближайшее время, т.е. при $t < T_h$, где T_h – временной горизонт (time horizon). Временной горизонт – значение времени, выбираемое исходя из динамических характеристик робота, траекторий препятствий, длительности расчёта объездных манёвров. В случае с транспортным модулем введение такого ограничения необязательно, поскольку СМБД отправляет информацию только о тех препятствиях, которые находятся в ближней зоне, определяемой параметрами СМБД.

Основной недостаток метода состоит в том, что не учитывается поведение препятствия, если оно является другим роботом или, например, человеком, который также стремится обойти препятствие. Для решения этой проблемы предложены алгоритмы построения RVO [64] и HRVO [110]. В нашей задаче такая проблема не ставится, однако при необходимости возможен переход к алгоритмам RVO и HRVO.

Управление движением транспортного модуля осуществляется на каждом временном интервале Δt , который задаётся исходя из вычислительных возможностей модуля и частоты получения новых сенсорных данных. Величина скорости, которую ТМ может достигнуть через интервал времени Δt , определяется максимально возможным ускорением модуля. При выборе новой скорости обычно учитывается это ограничение [76; 77]. Скорость

$\mathbf{v}_r(t + \Delta t) = (v_{rx}(t + \Delta t), v_{ry}(t + \Delta t))^T$ на следующем интервале Δt должна удовлетворять условию:

$$\begin{aligned} v_{rx}(t + \Delta t) &\in [v_{rx}(t) - \dot{v}_{rx}\Delta t, v_{rx}(t) + \dot{v}_{rx}\Delta t], \\ v_{ry}(t + \Delta t) &\in [v_{ry}(t) - \dot{v}_{ry}\Delta t, v_{ry}(t) + \dot{v}_{ry}\Delta t], \end{aligned} \quad (3.4)$$

здесь $\dot{v}_{rx}, \dot{v}_{ry}$ – ускорения вдоль осей системы координат, связанной с ТМ.

Это условие можно графически представить прямоугольной областью достижимых скоростей (RV). Данная область в общем случае может пересекаться с конусами $VO_{A|B_i}$; часть достижимой зоны, не принадлежащая VO , является множеством *достижимых скоростей, обеспечивающих объезд подвижных препятствий*. В статье [76] это множество обозначается как RAV :

$$RAV(t + \Delta t) = RV(t + \Delta t) \setminus VO_{A|B_1, B_2, \dots, B_m}(t). \quad (3.5)$$

Любая скорость из RAV обеспечивает объезд препятствия, однако ТМ должен выбирать такую скорость, которая близка к предпочтительной скорости $\mathbf{v}_r^{pref}$. Эта скорость, с которой ТМ должен двигаться к целевой точке, если препятствия отсутствуют. Например, она может определяться согласно пропорциональному закону, определяемому формулой (3.3).

Выбор скорости на следующем временном шаге является типовой задачей оптимизации, которая может быть записана следующим образом [85]:

$$\mathbf{v}_r^{new} = \arg \min_{\mathbf{v}_r \in RAV(t + \Delta t)} \|\mathbf{v}_r - \mathbf{v}_r^{pref}\|. \quad (3.6)$$

Это условие учитывает только динамические препятствия, однако, как было отмечено выше, на пути робота могут находиться статические препятствия. Обычно для статических препятствий тоже строятся области VO , но для препятствий произвольной формы это сделать затруднительно. Поэтому в данной работе из области RAV исключаются все те скорости, которые приведут к столкновению со статическим препятствием в момент времени $t + \Delta t$.

Проверка столкновения со статическим препятствием осуществляется следующим образом (рисунок 3.11).

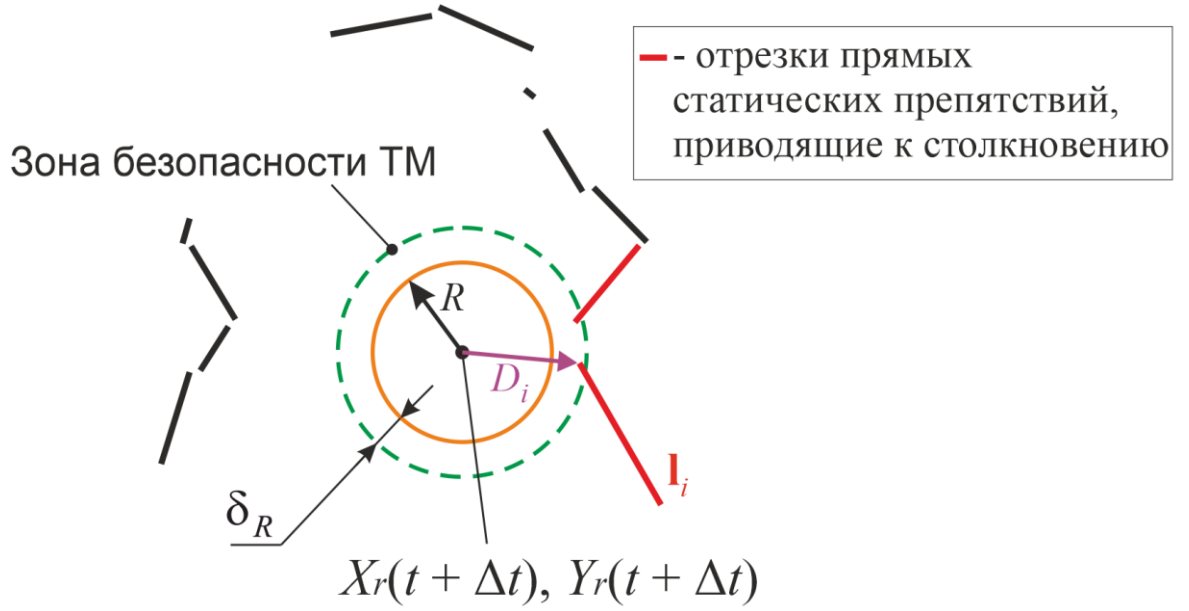


Рисунок 3.11. Проверка отрезков прямых, принадлежащих неподвижным препятствиям в ближней зоне робота

Пусть в момент времени $t + \Delta t$ робот находится в положении с координатами $X_r(t + \Delta t), Y_r(t + \Delta t)$. В этом положении проверяется, пересекает ли зону безопасности транспортного модуля, которая является окружностью с радиусом $R + \delta_R$, хотя бы один отрезок прямой l_i ($i = 1, 2, \dots, n$), принадлежащий статическому препятствию, δ_R – запас по расстоянию. То есть *столкновение с препятствием произойдет*, если хотя бы для одного отрезка прямой l_i выполняется условие:

$$D_i \leq R + \delta_R,$$

где D_i – минимальное расстояние от центра робота до i -го отрезка прямой l_i .

Исключая из области RAV все скорости $\mathbf{v}_r$, приводящие к столкновению МР со статическими препятствиями, получим область RAV^* . Тогда окончательно условие (3.6) можно переписать:

$$\mathbf{v}_r^{new} = \arg \min_{\mathbf{v}_r \in RAV^*(t+\Delta t)} \|\mathbf{v}_r - \mathbf{v}_r^{pref}\|. \quad (3.7)$$

Таким образом, алгоритм движения транспортного модуля в среде с подвижными и неподвижными препятствиями с помощью метода VO формулируется следующим образом:

Алгоритм 3.1.

1. **Входные параметры алгоритма:** список подвижных препятствий $\{\mathbf{O}_1, \mathbf{O}_2, \dots, \mathbf{O}_m\}$, отрезки прямых статических препятствий $\{\mathbf{l}_1, \mathbf{l}_2, \dots, \mathbf{l}_n\}$.
2. Рассчитать предпочтительную скорость $\mathbf{v}_r^{pref}$ транспортного модуля.
3. Построить область достижимых скоростей RV .
4. Для всех $\mathbf{O}_i$ из $\{\mathbf{O}_1, \mathbf{O}_2, \dots, \mathbf{O}_m\}$:
5. Построить $VO_{A|\mathbf{O}_i}$.
6. Построение RAV : исключить из RV все области $VO_{A|\mathbf{O}_i}$, $i = 1, 2, \dots, m$.
7. Построить RAV^* (используя параметры отрезков $\{\mathbf{l}_1, \mathbf{l}_2, \dots, \mathbf{l}_n\}$).
8. Если область RAV^* пуста, то:
9. Остановить ТМ: $\mathbf{v}_r^{new} = (0, 0)^T$.
10. Иначе:
11. Вычислить новую скорость $\mathbf{v}_r^{new} \in RAV^*$ по формуле (3.7).
12. Вычислить требуемые скорости колёс ТМ по формуле (3.1).

Если динамических препятствий нет, то шаги 4-6 не выполняются, и новая скорость выбирается из подмножества RV^* , которая исключает столкновение со статическими препятствиями.

Если область RAV^* пуста (условие на шаге 8), то это означает, что любой вектор скорости транспортного модуля может привести к столкновению в ближайшее время. В этом случае ТМ должен остановиться (шаг 9).

Для использования алгоритма в компьютерной модели и в реальной управляющей программе транспортного модуля область RV представляется как конечный набор векторов достижимых скоростей с заданным шагом дискретизации (рисунок 3.12). Соответственно область RAV^* является конечным подмножеством области RV .

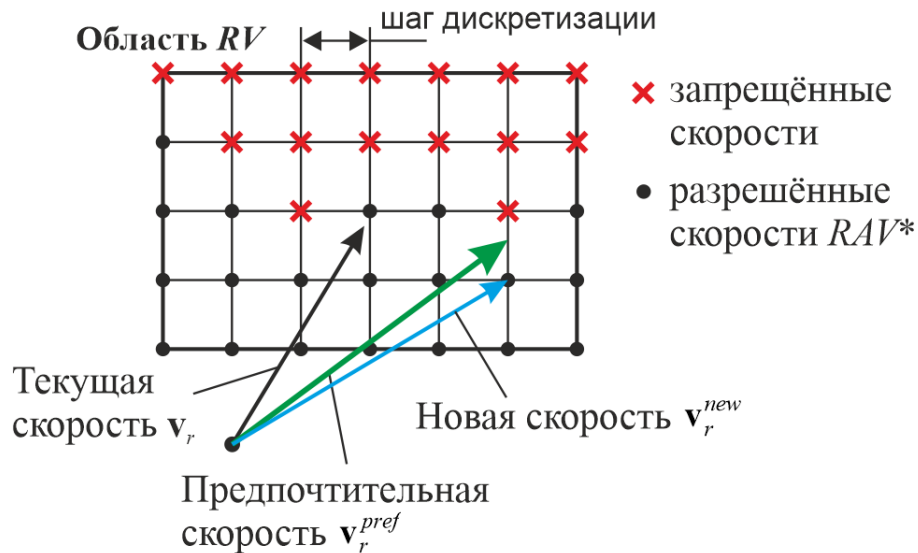


Рисунок 3.12. Конечный набор возможных скоростей RAV^*

Чем меньше шаг дискретизации, тем более плавным и точным будет движение ТМ. Однако с уменьшением шага (т.е. с увеличением размера набора скоростей), возрастает длительность расчёта алгоритма 3.1, что показано в эксперименте в главе 5.

3.5 Требования к межмодульным интерфейсам ТМ и коммуникационным каналам

Транспортный модуль имеет два основных межмодульных интерфейса:

- ТМ $\Leftrightarrow$ МИУ,
- ТМ $\Leftrightarrow$ СМБД.

У транспортного модуля также есть интерфейс для информационного взаимодействия с силовым модулем, однако в диссертации этот интерфейс не рассматривается, поскольку требования к его пропускной способности минимальные (с его помощью передаются только низкочастотные сервисные сообщения с частотой не более 1 Гц).

Необходимо определить требования к интерфейсам, межмодульным коммуникационным каналам (минимальная скорость передачи данных,

направление передачи) и определить топологию сети, состоящей из модулей мобильного робота.

Интерфейс ТМ $\Leftrightarrow$ МИУ определяется структурами данных, которые передаются между МИУ и ТМ (см. таблицу 2.2 и 2.3). Максимальный объём сообщения-задания, отправляемого из МИУ в ТМ (приём данных ТМ), будет составлять 21 байт:

- 1 – координата X_i целевой точки – 4 байта (тип float),
- 2 – координата Y_i целевой точки – 4 байта (тип float),
- 3 – координата X_0 исходной точки – 4 байта (тип float),
- 4 – координата Y_0 исходной точки – 4 байта (тип float),
- 5 – допустимое отклонение – 4 байта (тип float),
- 6 – предельное время прибытия – 1 байт (тип uint8).

Рассчитаем *минимальную* требуемую скорость передачи данных по интерфейсу ТМ $\Leftrightarrow$ МИУ (без учёта служебных данных протокола коммуникации).

Сообщение с заданием отправляется в начале перемещения ТМ и каждый раз, когда ТМ достигает очередной целевой точки с заданной точностью (подсостояние $S_{M.1.1.2}$ конечного автомата МИУ). Период отправки зависит от скорости перемещения ТМ и количества подвижных препятствий на пути робота к цели, т.е. от времени, затрачиваемого на достижение очередной целевой позиции.

Для расчёта периода отправки предположим, что модульный робот перемещается от точки к точке с максимальной скоростью 2 м/с по прямолинейной траектории и на его пути отсутствуют препятствия. Согласно размеру ближней зоны принимаем, что расстояния между целевыми точками одинаковы и равны 3 м. Тогда робот будет добираться до очередной точки за 1.5 с. В этом случае МИУ будет отправлять сообщение-задание с частотой $1/1.5 = 0.67$ Гц. Если скорость ТМ будет больше, то и частота отправки увеличится, но даже при скорости ТМ в 5 м/с частота передачи сообщений будет меньше 2 Гц. При частоте 2 Гц минимальная требуемая скорость передачи данных будет составлять 42 байт/с или 336 бит/с.

Объём данных, передаваемых ТМ в МИУ (отправка данных ТМ), составляет 9 байт:

- 1 – координата X_r текущей позиции – 4 байта (тип float),
- 2 – координата Y_r текущей позиции – 4 байта (тип float),
- 3 – оценка выполнения задачи – 1 байт (тип uint8).

Частота передачи сообщения-отчёта равна частоте передачи сообщения-задания. Минимальная скорость передачи данных при частоте 2 Гц будет составлять 18 байт/с или 144 бит/с.

Интерфейс ТМ $\Leftrightarrow$ СМБД определяется структурами данных, которые передаются между ТМ и СМБД (см. таблицы 2.4 и 2.5). Во второй главе было указано, что в одном сообщении СМБД могут содержаться данные о 100-150 отрезках прямых, которые описывают статические препятствия. Упорядоченный набор параметров i -го отрезка прямой имеет объём 16 байт:

- 1 – x_{i1}, y_{i1} – координаты первого конца отрезка – 8 байт (тип float),
- 2 – x_{i2}, y_{i2} – координаты второго конца отрезка – 8 байт (тип float).

Соответственно набор параметров 150 отрезков прямых будет занимать 2400 байт.

Объём данных, требуемый для передачи информации о подвижных препятствиях, напрямую зависит от количества препятствий в ближней зоне. Принимаем, что модульный робот находится в такой среде, где максимальное количество подвижных препятствий в ближней зоне не превышает 10 (см. раздел 2.4). Упорядоченный набор параметров i -го препятствия будет занимать 20 байт в сообщении:

- 1 – x_{Oi}, y_{Oi} – координаты центра габаритной окружности препятствия – 8 байт (тип float),
- 2 – v_{xi}, v_{yi} – проекции вектора скорости препятствия на оси базовой системы координат – 8 байт (тип float),
- 3 – радиус габаритной окружности препятствия r_{Oi} – 4 байта (тип float),

Соответственно объём данных о 10 препятствиях будет равен 200 байт.

Всего объём одного сообщения отправляемого СМБД в ТМ (приём данных) будет составлять 2608 байт при принятых параметрах:

- координаты текущей позиции (X_r , Y_r) – 8 байт (тип float),
- набор отрезков прямых статических препятствий – 2400 байт,
- набор данных о подвижных препятствиях – 200 байт.

Видно, что более 92% объёма полезной нагрузки одного сообщения занимает информация о параметрах отрезков статических препятствий.

Сенсорный модуль ближнего действия должен отправлять сообщение объёмом 2608 байт с частотой *не менее* 20 Гц согласно принятым условиям движения, которые рассмотрены в конце раздела 2.4. Тогда минимальная скорость передачи данных по интерфейсу ТМ $\Leftrightarrow$ СМБД будет составлять 52160 байт/с или 417.3 кбит/с.

Объём данных, передаваемых ТМ в СМБД, составляет 16 байт:

- 1 – координата X_r текущей позиции – 4 байта (тип float),
- 2 – координата Y_r текущей позиции – 4 байта (тип float),
- 3 – скорость ТМ в направлении оси x – 4 байта (тип float),
- 4 – скорость ТМ в направлении оси y – 4 байта (тип float).

Частота передачи данных транспортным модулем в СМБД должна быть равна частоте приёма сообщений (не менее 20 Гц) для синхронизации алгоритма локализации в СМБД, поскольку сенсорные системы используют данные одометров для расчёта координат робота в среде. Тогда минимальная требуемая скорость передачи (отправка данных транспортным модулем) будет равна 320 байт/с или 2.56 кбит/с.

Проведённый анализ интерфейсов позволяет выбрать подходящую коммуникационную сеть для связи между модулями. Скорости передачи данных интерфейсов ТМ $\Leftrightarrow$ МИУ и ТМ $\Leftrightarrow$ СМБД значительно отличаются (таблица 3.1), но канал связи должен быть унифицированным для обеспечения взаимозаменяемости модулей. Тогда выбор канала связи будет определяться интерфейсом ТМ $\Leftrightarrow$ СМБД, как самым загруженным в системе.

Таблица 3.1. Основные требования к интерфейсам ТМ

Интерфейс		Минимальная скорость передачи, кбит/с	Частота передачи, Гц
ТМ $\Leftrightarrow$ МИУ	Приём	0.336	2
	Отправка	0.144	2
ТМ $\Leftrightarrow$ СМБД	Приём	417.3	20
	Отправка	2.56	20

Рассмотрим требования к сети из *полнофункциональных* модулей. Учитывая обобщённую иерархическую структуру системы управления модульного робота сеть должна допускать непосредственное взаимодействие между ТМ, МИУ и СМБД. В этом случае возможны следующие топологии сети:

- полносвязная,
- ячеистая,
- звезда,
- шина.

В статье [8] показано, что для полнофункциональных модулей допустимо использование топологий сети «шина» и «звезда». Однако сети с топологией «звезда» обеспечивают более широкополосный канал, что необходимо для нагруженных интерфейсов сенсорных модулей. Шинная топология предпочтительна для передачи команд исполнительного уровня управления [6], поэтому она широко распространена в гомогенных модульных роботах.

Канал связи должен обеспечивать передачу данных в дуплексном режиме со скоростью (*пропускной способностью*) не менее 835 кбит/с. Запас взят из-за того, что при расчёте скорости передачи не учитывались *служебные данные*. При этом канал может быть и полудуплексным, но тогда скорость передачи данных должна быть в два раза больше. Канал межмодульной связи должен быть проводным.

Протокол коммуникации не должен назначать в сети ведущий узел (мастер), поскольку каждый модуль может инициировать передачу данных. Для

организации сети из полнофункциональных модулей с топологией типа «шина» хорошо подходят стандарты CAN (CAN FD) и Ethernet.

На рисунке 3.13 показана схема подключения модулей к шине CAN.

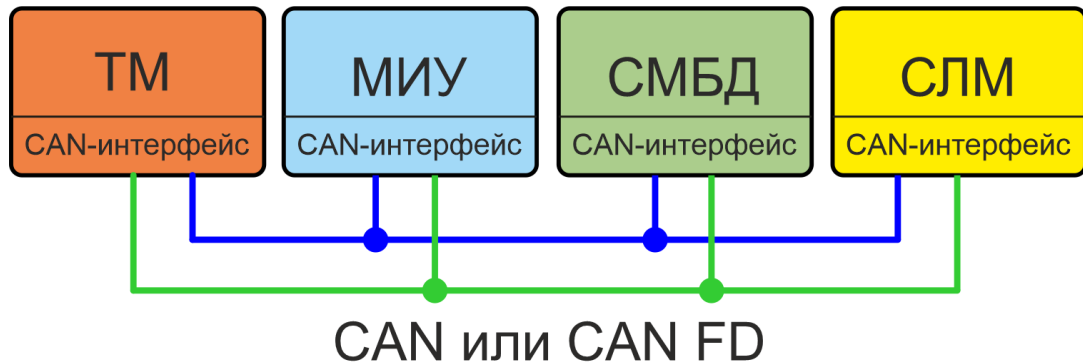


Рисунок 3.13. Схема подключения полнофункциональных модулей к шине CAN

Основные преимущества шины CAN: обмен сообщениями в реальном времени, простота реализации, высокая помехоустойчивость, обнаружение коллизий сообщений и контроль ошибок. По сравнению с классическим CAN версии 2.0В стандарт CAN FD позволяет передавать до 64 байт полезной нагрузки в одном кадре с битовой скоростью до 8 Мбит/с.

Стандарт Ethernet широко распространён, обеспечивает высокие скорости передачи данных (от 100 Мбит/с) и даёт возможность использовать известные интернет-протоколы (TCP/IP, UDP).

По сравнению с CAN технология Ethernet является более дорогостоящей и требовательной к вычислительным ресурсам встраиваемых систем. При этом кабели Ethernet подвержены наводкам от силового оборудования, которое в случае мобильной робототехники может быть достаточно мощным (электроприводы). Однако сетевые протоколы на базе Ethernet обеспечивают шифрование трафика, в отличие от шины CAN, которая практически не защищена от несанкционированного доступа к данным, но в пределах конструкции МР такая защита не требуется. С другой стороны, электрические соединения для стандарта Ethernet требуют многоконтактных разъёмов, что приводит к определённым трудностям при разработке конструкции МР и обеспечении их надёжности и защищённости от воздействий внешней агрессивной среды. В то же время шина

CAN давно используется в транспортных средствах (автомобили, сельскохозяйственные машины) и обеспечивает надёжный коммуникационный канал для электронных компонентов в современных автомобилях. Тем не менее, следует отметить, что в настоящее время намечается тенденция на вытеснение CAN даже из систем коммуникаций транспортных средств [43].

Использование такого популярного в компьютерной технике интерфейса как USB в рассматриваемой модульной архитектуре затруднено. Это обусловлено тем, что устройства USB работают в режиме «ведущий-ведомый», что противоречит нашей концепции: каждый модуль может инициировать передачу данных в *любой момент времени* (например, в случае экстренной ситуации).

3.6 Реализация распределённых вычислений на основе модульной и субмодульной архитектуры СУ МР

Разработанный принцип полной функциональности позволяет разделить вычислительную нагрузку между МИУ, ТМ и СМБД, что было обосновано во второй главе. Однако вычислительная сложность алгоритмов СУ ТМ, рассмотренных в разделах 3.1 – 3.4, требует использования в СУ этого модуля, как минимум, одноплатных компьютеров, поскольку помимо выполнения поведенческих задач ТМ также управляет приводами колёсной платформы.

Для решения данной проблемы предлагается рассматривать ТМ как *функционально-модульную архитектуру с узлами-субмодулями* [58], в которой вычисления распределяются между встраиваемыми системами – микроконтроллерами невысокой производительности. Такая архитектура позволит распределить вычислительную нагрузку различной функциональной значимости между субмодулями, особенно в тех случаях, когда число приводов и датчиков может увеличиваться или их тип может изменяться. Так, транспортный модуль с всенаправленным мобильным механизмом имеет четыре ведущих механум-колеса.

При организации распределённых вычислений появляется возможность оперативно изменять конфигурацию ТМ за счёт комбинирования субмодулей.

Предлагается следующая схема распределённой архитектуры системы управления полнофункционального колёсного ТМ (рисунок 3.14).

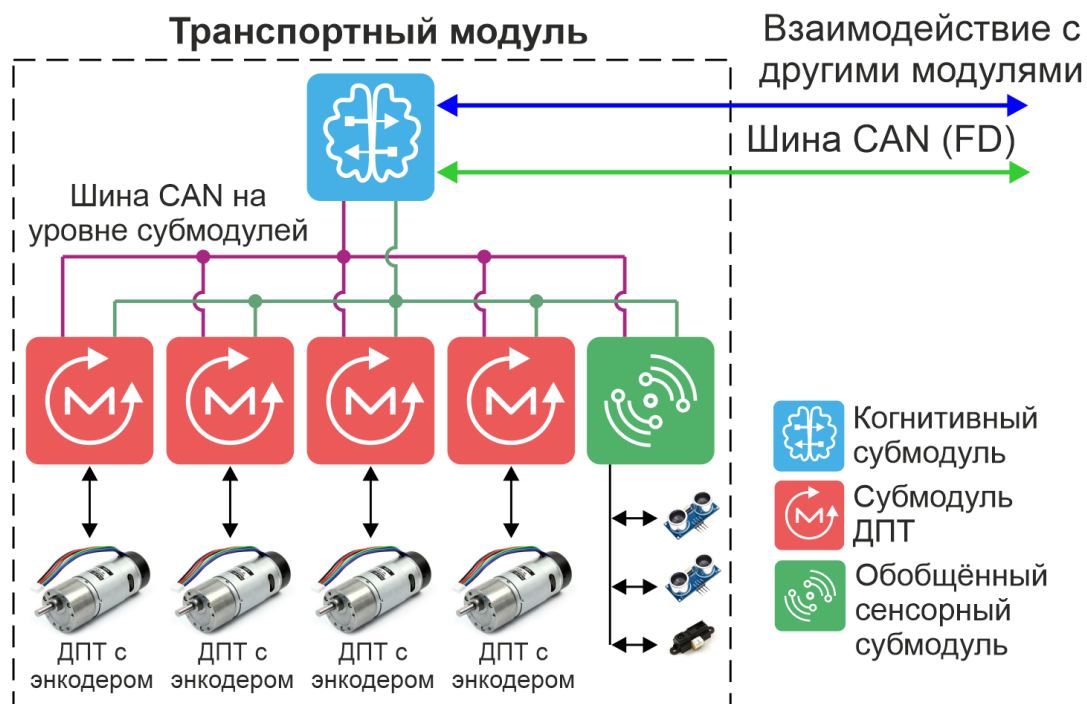


Рисунок 3.14. Распределённая архитектура системы управления транспортного модуля

Структура транспортного модуля основана на трёх типах субмодулей:

1. *Когнитивный субмодуль* – предназначен для реализации алгоритма объезда препятствий, взаимодействия с остальными полнофункциональными модулями по межмодульным интерфейсам (как показано в предыдущем разделе), а также для выдачи задающих воздействий субмодулям. Когнитивный субмодуль является супервизором (мастером) по отношению к другим субмодулям.

2. *Субмодуль-привод* – управляет работой подключаемого к нему электродвигателя (на рисунке 3.14 представлен частный случай – субмодуль для управления ДПТ).

3. *Обобщённый сенсорный субмодуль* – позволяет подключать различные экстероцептивные (external state) и проприоцептивные (internal state) датчики: дистанционные датчики расстояний, гироскопы, акселерометры и т.д.

Наличие обобщённого сенсорного субмодуля в транспортном модуле является опциональным; он может быть использован в тех ситуациях, когда СМБД (на верхнем уровне иерархии) отсутствует в мобильном роботе. Его наличие в субмодуле может быть вызвано также особенностями применения МР.

Для каждого субмодуля, как и для всех полнофункциональных модулей, должен соблюдаться *принцип полной функциональности*. Субмодуль должен иметь собственную систему управления, базирующуюся на микроконтроллере невысокой производительности.

Информационное взаимодействие между субмодулями должно осуществляться в «жёстком» реальном времени (например, шина CAN или RS-485). На уровне субмодулей частота передачи данных может достигать нескольких десятков герц, в зависимости от быстродействия системы. Сеть-шина, к которой подключены субмодули, должна допускать асинхронную работу, если частоты передачи сообщений узлами сети будут отличаться.

Рассмотрим подробнее устройство субмодулей.

Система управления *когнитивного субмодуля* отвечает за реализацию алгоритма объезда подвижных/неподвижных препятствий (алгоритм 3.1), а также за формирование вектора заданных скоростей колёс в соответствии с формулой (3.1) при известном векторе $\mathbf{v}_r^{new}$:

$$\begin{pmatrix} \dot{\phi}_1 \\ \dot{\phi}_2 \\ \dot{\phi}_3 \\ \dot{\phi}_4 \end{pmatrix} = \mathbf{J} \mathbf{v}_r^{new} = \frac{1}{R+r} \begin{pmatrix} 1 & -1 & -(l_x + l_y) \\ 1 & 1 & l_x + l_y \\ 1 & 1 & -(l_x + l_y) \\ 1 & -1 & l_x + l_y \end{pmatrix} \cdot \begin{pmatrix} v_{rx}^{new} \\ v_{ry}^{new} \\ 0 \end{pmatrix}.$$

При этом требуемая угловая скорость i -го механум-колеса должна отрабатываться системой управления соответствующего субмодуля-привода с помощью цифрового регулятора скорости (например, ПИ-регулятора скорости).

Когнитивный субмодуль помимо указанных функций исполняет роль «моста» между компьютером (ПК) высокого уровня и другими субмодулями (рисунок 3.15). Это необходимо для первичной настройки субмодулей: установки

коэффициентов ПИД-регуляторов, параметров фильтрации и интерполяции и т.д. Подключение этого субмодуля к компьютеру может осуществляться по проводной (USB) и беспроводной связи (Wi-Fi).

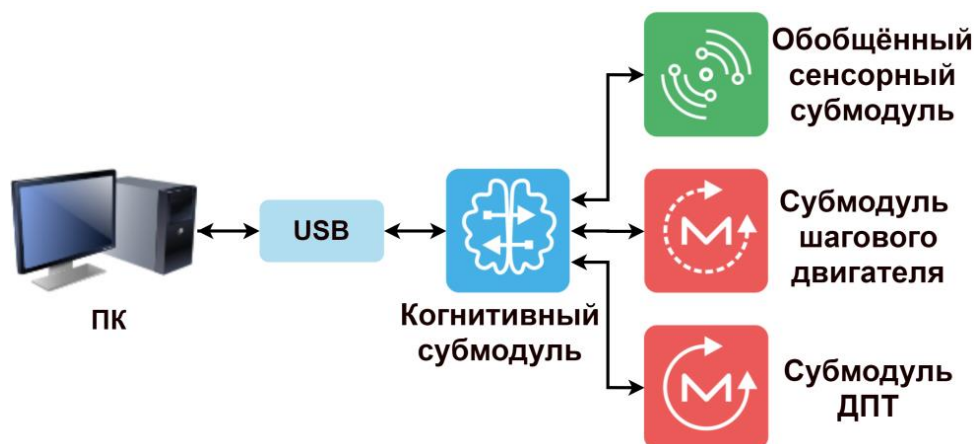


Рисунок 3.15. Настройка субмодулей с помощью компьютера

Субмодуль-привод – электронный контроллер, управляющий различными электродвигателями: коллекторными, бесколлекторными ДПТ, шаговыми двигателями, сервомоторами. Основные функции субмодуля: управление валом по скорости, положению и/или моменту. В зависимости от типа регулируемого электродвигателя субмодули-приводы могут быть четырёх основных видов: контроллер коллекторного ДПТ, контроллер бесколлекторного ДПТ, контроллер шагового двигателя, контроллер сервомоторов.

Обобщённый сенсорный субмодуль – электронное устройство, предназначенное для подключения различных датчиков, преимущественно простых датчиков, энкодеров, концевых выключателей, а также инерционных датчиков (гироскопов, акселерометров). Основные функции субмодуля: сбор и обработка сенсорных данных, комплексирование. Предполагается, что к одному сенсорному субмодулю может быть подключено не более четырёх датчиков.

Таким образом, система управления всего модульного робота представляет собой иерархическую модульную архитектуру с тремя уровнями управления (рисунок 3.16). На первом уровне взаимодействие происходит между субмодулями, на втором – между полнофункциональными модулями, на третьем уровне располагается МИУ, как главный узел модульной системы.

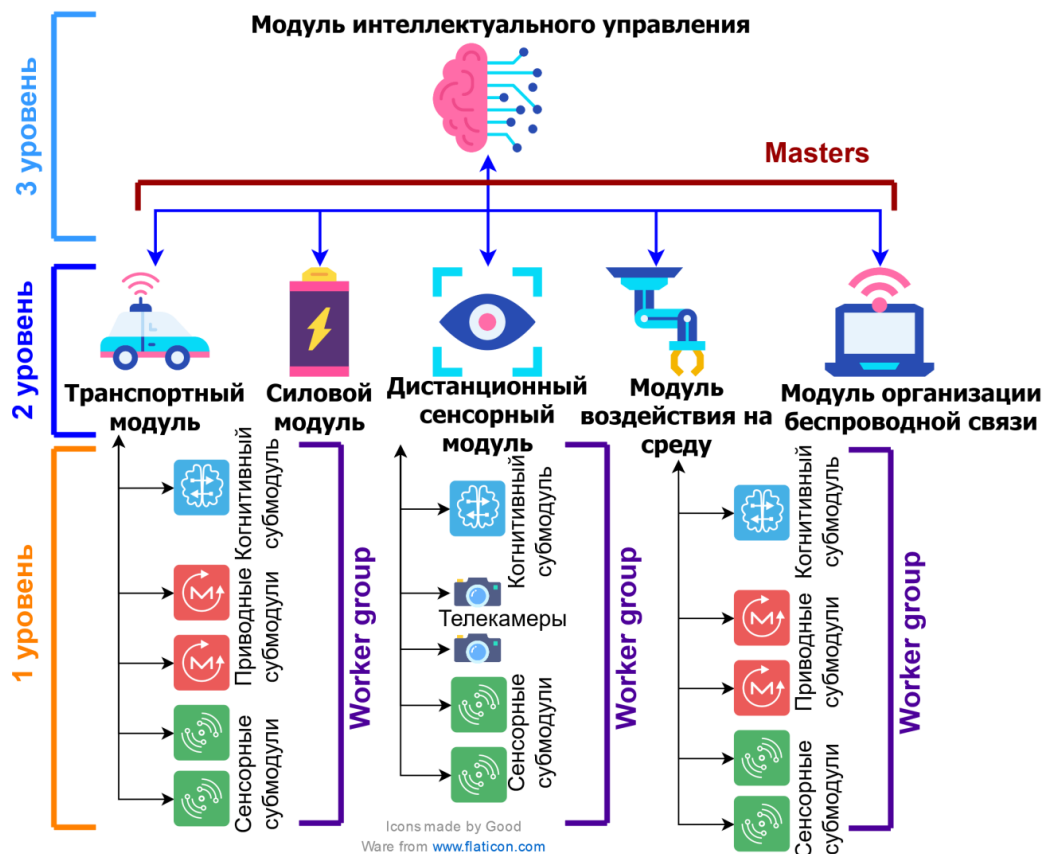


Рисунок 3.16. Иерархическая структура СУ модульного мобильного робота

На схеме видно, что другие полнофункциональные модули тоже могут быть построены на основе субмодулей.

Модули второго уровня составляют группу Masters для субмодулей первого уровня – Workers, при этом выполняется взаимодействие по принципу «Master-Worker group». На уровне субмодулей также реализуется оперативная реконфигурация посредством использования соответствующих драйверов и оказывается возможным использование разработок субмодулей сторонних производителей.

Оценим объём передачи сообщений между субмодулями в ТМ. Для этого необходимо определить, какие типовые данные будут отправлять субмодули при работе ТМ. При этом низкочастотные служебные сообщения приниматься в расчёт не будут (с частотой менее 10 Гц).

Рассматриваемый транспортный модуль состоит из платформы с четырьмя механум-колёсами (раздел 3.2). Тогда ТМ должен включать в себя следующие субмодули:

- один когнитивный субмодуль,
- четыре субмодуля-привода для управления колёсами,
- опционально один обобщённый сенсорный субмодуль.

Следует отметить, что в общем случае количество субмодулей-приводов может быть больше или меньше в зависимости от конструкции шасси ТМ.

Управляет всеми субмодулями-приводами когнитивный субмодуль, задающий требуемые скорости и/или положения валов электродвигателей. Сообщение с заданной скоростью/положением для одного субмодуля-привода имеет следующую структуру:

- заданная угловая скорость – 4 байта (float),
- заданное угловое положение – 4 байта (float).

Это сообщение должно отправляться для каждого субмодуля-привода, либо в одном сообщении должны быть заданы скорости и положения для всех субмодулей одновременно.

Частота отправки сообщений с задающими скоростями/положениями зависит от быстродействия ТМ и от частоты работы контура управления, образованного модулем интеллектуального управления, транспортным модулем и сенсорным модулем ближнего действия. Частота работы этого контура определяется частотой отправки данных сенсорным модулем. В данном случае она должна быть не меньше 20 Гц. При этом частота работы замкнутых контуров в субмодулях, которые управляют двигателями, будет выше (например, 100 Гц).

Каждый субмодуль-привод должен отправлять фактические значения скорости/положения:

- фактическая скорость вращения вала – 4 байта (float),
- фактическое положение вала – 4 байта (float),
- метка времени – 7 байт (uint56).

Метка времени – момент времени в микросекундах, в который субмодуль-привод отправляет сообщение на шину.

Частота отправки сообщения должна быть равна частоте приёма сообщения-задания от когнитивного субмодуля (в данном случае 20 Гц).

К обобщённому сенсорному субмодулю, как было отмечено выше, могут быть подключены различные датчики в количестве до 4 шт. Будем считать, что каждый датчик возвращает данные в формате числа с запятой (4 байт), тогда объём одного сообщения будет равен 16 байт.

Минимальная требуемая пропускная способность шины для субмодулей ТМ без учёта служебных данных протокола рассчитывается следующим образом:

$$V = 8f(nC_k + nC_{np} + C_c), [\text{бит/с}],$$

где f – частота публикации сообщений на шину, Гц,

n – количество субмодулей-приводов,

C_k – объём сообщения когнитивного субмодуля (8 байт),

C_{np} – объём сообщения субмодуля-привода (15 байт),

C_c – объём сообщения обобщённого сенсорного субмодуля (16 байт).

На рисунке 3.17 показаны графики зависимости требуемой пропускной способности от частоты публикации в диапазоне от 10 до 500 Гц и от количества субмодулей-приводов (2, 3, 4, 6 и 8).

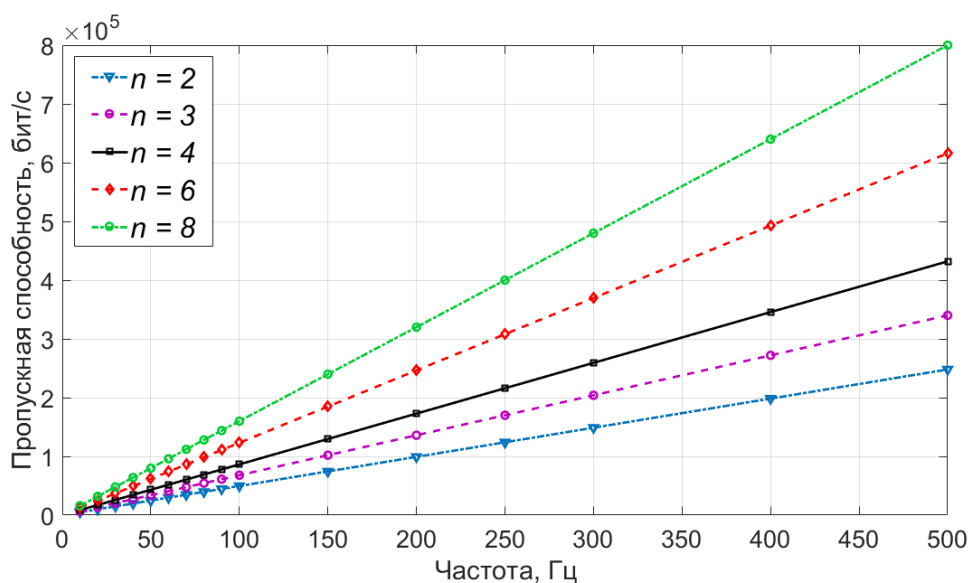


Рисунок 3.17. Зависимость требуемой пропускной способности от частоты публикации и количества субмодулей-приводов

Из графиков видно, что максимальная требуемая пропускная способность без учёта служебных данных не превышает 800 кбит/с при частоте публикации 500 Гц в случае подключения 8-ми субмодулей-приводов. Для рассматриваемого случая четырёх субмодулей-приводов пропускная способность не превышает 432 кбит/с.

Таким образом, требуемая пропускная способность должна быть не менее 800 кбит/с. В качестве коммуникационной сети для субмодулей хорошо подходит шина CAN, обеспечивающая высокую пропускную способность и работу в реальном времени.

Учёт служебных данных для более точного расчёта требуемой пропускной способности возможен только при известном протоколе межмодульной коммуникации. Выбор протокола и уточнённый расчёт рассматриваются в следующей главе.

3.7 Выводы по третьей главе

Были разработаны диаграммы состояний для конечных автоматов МИУ и ТМ с применением языка UML. Диаграммы состояний позволяют формально описать взаимодействие МИУ и ТМ с учётом внутренних и внешних событий (истечение времени, неисправность, столкновение и т.д.). Определены все основные условия переходов между состояниями и подсостояниями. Основное преимущество диаграмм состояний UML – возможность легко перейти к программной реализации автоматов.

В данной диссертации приведён способ формализации взаимодействия модулей, что недостаточно подробно рассмотрено в описаниях модульных роботов, обзор которых приведён в первой главе. Это даёт теоретическое обоснование для разработки компьютерной модели робота.

Для транспортного модуля, представляющего собой мобильную платформу с четырьмя механум-колёсами, был разработан алгоритм движения к точке с

объездом подвижных и неподвижных препятствий. Для объезда подвижных препятствий используется популярный алгоритм «Velocity Obstacle» [76], который целесообразно применять, если подвижные препятствия представляются в виде окружностей. При этом, согласно принципу полной функциональности транспортного модуля, алгоритм объезда препятствий может быть любым, если получаемые от СМБД данные достаточны для его работы.

Было показано, что коммуникационная сеть, включающая в себя МИУ, ТМ и СМБД, должна иметь топологию типа «шина»; пропускная способность сети должна быть не менее 835 кбит/с. Требования к сети определяются самым загруженным интерфейсом ТМ $\leftrightarrow$ СМБД из-за значительного потока сенсорных данных. В коммуникационной сети не должно быть ведущего узла (мастера). Показано, что в качестве коммуникационной сети следует использовать шину CAN FD.

Предложено рассматривать ТМ как модульную многоуровневую архитектуру с узлами-субмодулями. Такой же подход применим и к другим полнофункциональным модулям. Тогда система управления всего модульного робота представляет собой иерархическую функционально-модульную архитектуру с тремя уровнями управления. Это позволяет распределить вычислительную нагрузку каждого из модулей второго уровня иерархии между его субмодулями, что даёт возможность строить их системы управления на базе относительно дешёвых вычислительных устройств невысокой производительности, имеющих малые массогабаритные характеристики и низкое энергопотребление.

Для модульной архитектуры транспортного модуля был определён состав субмодулей и их функционал. Определены сообщения, которые субмодули должны отправлять на общую шину CAN. Показано, что требуемая пропускная способность шины должна быть не менее 800 кбит/с.

ГЛАВА 4. МЕЖМОДУЛЬНОЕ ИНФОРМАЦИОННОЕ ВЗАИМОДЕЙСТВИЕ

В предыдущей главе было показано, что в качестве коммуникационной сети для модулей и субмодулей подходит шина CAN. При этом на уровне полнофункциональных модулей вместо классической шины CAN следует использовать модификацию CAN FD. Однако шина CAN реализует только первый и второй уровни сетевой модели OSI.

В этой главе описывается выбор протокола коммуникации для шины CAN с целью реализации остальных уровней модели OSI. Сначала рассматривается протокол ModRob [7], который разрабатывался автором совместно с коллегами. Протокол позволял работать субмодулям в реальном времени, однако имел определённые недостатки. В связи с этим было решено использовать протокол OpenCyphal, зарекомендовавший себя в сфере беспилотных летательных аппаратов [42], где вычислительные узлы работают в жёстком реальном времени.

С учётом спецификации протокола OpenCyphal рассмотренные во второй главе структуры данных для межмодульного взаимодействия были записаны на специальном языке DSDL. Эти типы (структуры) данных можно легко конвертировать в программный код для непосредственного использования в управляющих программах модулей/субмодулей.

Для уровня модулей и субмодулей были определены сообщения, которые должны передаваться во время работы системы. С учётом принятых исходных данных (количество модулей/субмодулей на шине, битовая скорость шины и т.д.) проводится анализ загруженности шины как на уровне полнофункциональных модулей, так и на уровне субмодулей транспортного модуля.

Протокол OpenCyphal позволяет подключать модули/субмодули работа в режиме «plug and play» благодаря использованию анонимных сообщений и специальных служебных типов данных. При этом в сети обязательно должен быть как минимум один узел, который имеет фиксированный идентификатор.

Использование соответствующих библиотек протокола OpenCyphal даёт возможность реализовать оперативное реконфигурирование СУ МР, что не удалось сделать средствами разработанного протокола ModRob.

4.1 Протокол информационного взаимодействия модулей и субмодулей робота

Полнофункциональные модули всего робота подключаются к одной шине CAN, а субмодули транспортного модуля – к другой (рисунок 4.1). При этом, если сенсорный модуль ближнего действия отправляет в ТМ большой объём информации (до 10 подвижных препятствий и до 150 отрезков прямых), то вместо классической шины CAN на уровне модулей должна использоваться шина CAN FD с большей пропускной способностью.

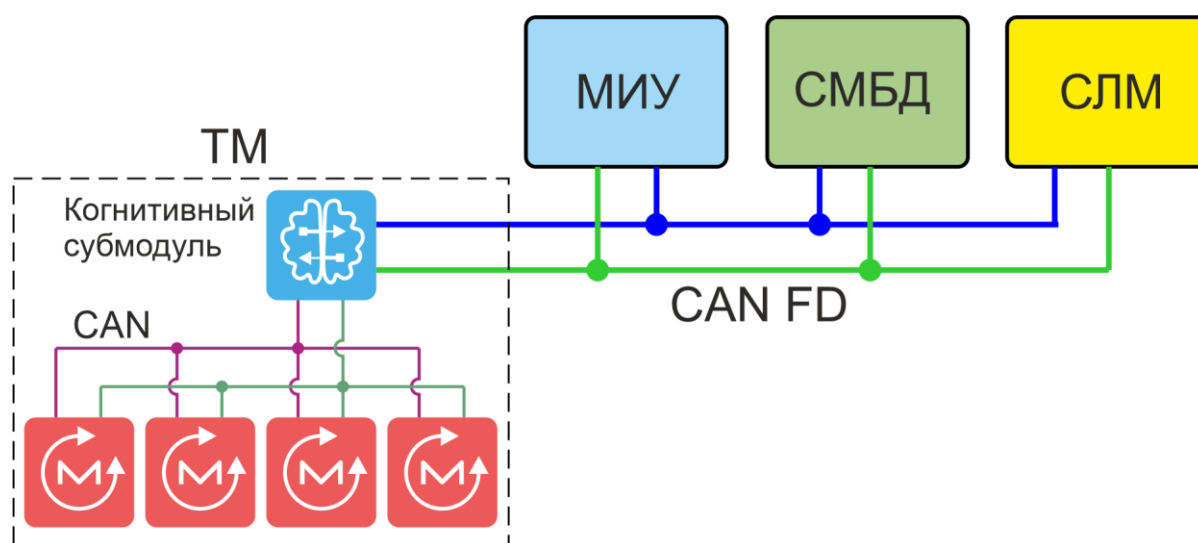


Рисунок 4.1. Использование двух шин CAN на двух уровнях иерархии модульной архитектуры СУ МР

На основе шины CAN есть готовые протоколы верхнего уровня. К наиболее известным из них относятся CANopen [66], SAE J1939 [45], CANaerospace [36] и OpenCyphal (прежнее название UAVCAN) [41].

Изначально автором совместно с коллегами разрабатывался собственный протокол под названием ModRob [7; 9; 55]. Протокол отличается минимализмом и

простотой использования: каждое сообщение занимает один кадр CAN, по шине передаются только числовые значения с плавающей запятой (по одному числу типа float на сообщение).

Каждый узел в сети ModRob имеет свой идентификатор (`moduleID`) и набор переменных со своими идентификаторами (`variableID`). Протокол основан на принципе «связывания» переменных: при изменении заданной переменной одного узла (модуля), переменные других модулей, подписанные на неё, также изменяются. Для этого сообщение, содержащее информацию о новом значении заданной переменной, должно публиковаться с некоторой периодичностью.

Программный узел может принимать и публиковать многоадресные сообщения. Каждое ModRob-сообщение занимает только один расширенный кадр CAN, как было отмечено выше. В таблице 4.1 представлены основные виды ModRob-сообщений.

Таблица 4.1. Основные виды сообщений в протоколе ModRob

Название команды	Входные параметры	Описание
Контрольный сигнал (Heartbeat)	ID субмодуля	Сообщение, используемое для синхронизации субмодулей на шине
Установка значения	1. ID целевого субмодуля 2. ID целевой переменной 3. Значение переменной	Установка нового значения целевой переменной целевого субмодуля
Установка частоты публикации	1. ID целевого субмодуля 2. ID целевой переменной 3. Значение частоты	Установка частоты публикации целевой переменной целевого субмодуля
Публикация переменной	1. ID субмодуля-отправителя 2. ID публикуемой переменной 3. Значение переменной	Публикация значения целевой переменной на общую шину (широковещательное сообщение)
Установка подписки	1. ID субмодуля-подписчика 2. ID переменной подписчика 3. ID модуля-отправителя 4. ID переменной отправителя	Команда подписывает (связывает) переменную субмодуля-подписчика на переменную субмодуля-отправителя

В работах [7; 55] подробно описан эксперимент, в котором исследовалась возможность использования протокола ModRob для межмодульного информационного взаимодействия субмодулей в реальном времени. Анализ

результатов эксперимента показал, что использование протокола ModRob позволяет осуществлять управление субмодулями в реальном времени – максимальное запаздывание реакции привода субмодуля не превышает 50 мс. При этом время, затрачиваемое на отправку сообщения по шине CAN, не превышает 16 мкс. Подробнее эксперимент рассматривается в пятой главе.

К недостаткам протокола можно отнести отсутствие поддержки многокадровых сообщений (что одновременно является и преимуществом), отсутствие приоритета сообщений и таких сервисов, как синхронизация времени узлов. Использование только однокадровых сообщений упрощает программную реализацию протокола и снижает требования к оперативной памяти вычислителя модуля. Однако это существенно усложняет передачу и приём структурированных данных большого объёма (например, сенсорной информации).

Необходимость перехода к готовому протоколу обусловлена тем, что такие протоколы как ModRob, разрабатываемые небольшим кругом лиц, не имеют широкой распространённости, по сравнению, например, с CANopen или SAE J1939. А это практически исключает возможность интеграции в нашу систему модулей *сторонних производителей*.

Разработка собственного протокола ModRob позволила определить требования к протоколу межмодульной коммуникации в иерархической архитектуре СУ, что, в свою очередь, обеспечило формализацию поиска и выбора протокола, соответствующего заданным требованиям, из числа признанных сообществом разработчиков систем управления. Были сформулированы следующие требования к протоколу межмодульной коммуникации [59]:

1. Транспорты³ протокола (в соответствии с моделью OSI) должны обеспечивать детерминированную доставку сообщений модулей и субмодулей со скоростью не менее 835 кбит/с (раздел 3.5).

³ Под «транспортом» здесь понимается часть спецификации протокола, которая определяет требования к первым пяти уровням протокола в соответствии с сетевой моделью OSI.

2. Протокол должен быть определён как стандарт с открытым исходным кодом.
3. Протокол должен иметь одну или несколько портативных (мультиплатформенных) открытых реализаций.
4. Желательно, чтобы протокол межмодульного взаимодействия мог быть реализован на основе разных транспортов: CAN, Ethernet, UART.
5. Протокол должен быть реализуем на встраиваемых вычислительных устройствах, и занимать минимальный объём памяти программ и оперативной памяти.
6. Протокол должен поддерживать режим «plug and play» и «горячее» подключение.
7. Сообщения в протоколе должны быть типизированными; протокол должен исключать приём и передачу некорректных сообщений.
8. Транспорты протокола не должны требовать наличие явного основного узла (мастера) в сети для передачи данных.
9. Протокол с транспортом, который способен предоставлять узлу информацию о топологии соединения других узлов в сети, должен иметь приоритет.
10. Должен существовать унифицированный способ передачи сообщений между различными транспортом протокола.
11. Как можно больше уровней (в смысле модели OSI) протоколов транспортов должно быть реализовано на аппаратном уровне.
12. Физический и/или канальный уровень транспортов протокола должны быть защищены от помех и/или иметь механизмы обнаружения и исправления ошибок.

Протокол OpenCyphal практически полностью соответствует этим требованиям [59].

Применение иных протоколов на шине CAN затруднено по различным причинам. Протокол CANopen, используемый в промышленности, транспорте, медицинской технике, фактически является закрытым стандартом (не

соответствует требованию №2). Для доступа к спецификации и другим документам нужно быть членом сообщества «CAN in Automation» (CiA) и оплачивать ежегодный взнос. Протокол SAE J1939, широко распространённый в транспортной технике (легковые автомобили, грузовики, сельскохозяйственные машины и др.), поддерживает скорость только до 500 Кбит/с, что недостаточно для коммуникации модулей и субмодулей (не соответствует требованию №1). Открытый протокол CANaerospace обладает меньшими возможностями по сравнению с OpenCyphal (например, нет готовых реализаций режима «plug and play»), хотя является достаточно распространённым. Также протокол CANaerospace не соответствует требованиям №4 и №10.

OpenCyphal – сетевой стандарт для организации работы в реальном времени распределённых вычислительных узлов. Протокол широко распространён в сообществе разработчиков беспилотных летательных аппаратов, а также в робототехнике. Основные преимущества протокола: обеспечение работы в жёстком реальном времени, возможность создания абстрактных межмодульных интерфейсов, обеспечение отказоустойчивости сети, возможность подключения модулей в режиме «plug and play». Стек протокола реализуется на широком спектре устройств: начиная от маломощных микроконтроллеров и заканчивая персональными компьютерами. OpenCyphal использует различные протоколы канального и физического уровня модели OSI: CAN, UART, Ethernet.

Протокол OpenCyphal является открытым стандартом (лицензия MIT); в сети Интернет легко найти готовые примеры для разработки совместимого с протоколом устройства.

Сеть OpenCyphal является *одноранговой сетью*, где каждый узел имеет собственный уникальный идентификатор (node-ID) в диапазоне от 0 до 127 (в шине CAN). Узлы сети осуществляют коммуникацию с помощью двух методов:

- *публикация сообщений* – основной метод обмена данными вида «один-ко-многим» со схемой «издатель-подписчик» (publisher/subscriber),

- *вызов сервисов* – метод обмена данными «один к одному» со схемой «запрос-ответ» (request/response).

Сообщения в сети объединяются в тематические группы (subjects) в зависимости от смыслового значения полезных данных (например, группа, включающая в себя только те сообщения, в которых содержится информация о скорости). Сообщения, принадлежащие одной группе, относятся к одной функции или процессу внутри системы. Запросы/ответы также объединяются в группы (сервисы).

Каждая тематическая группа сообщений имеет свой уникальный идентификатор (subject-ID), используя который узлы подписываются на сообщения. Каждый сервис также имеет свой уникальный идентификатор (service-ID). В спецификации OpenCyphal эти два вида идентификаторов имеют общий термин – port-ID⁴.

Для каждого метода коммуникации используются типы данных, определяемые специальным языком DSDL (data structure description language), также разработанным сообществом OpenCyphal. Каждый тип данных имеет своё имя, версию, атрибуты и, опционально, фиксированный идентификатор (port-ID).

Для совместимости типов данных в сети OpenCyphal разработчики должны руководствоваться требованиями и рекомендациями, изложенными в спецификации [41]. Каждый разработчик должен создавать свои типы данных на основе готового набора стандартных типов от сообщества OpenCyphal. Например, режим работы узла описывается таким стандартным типом данных (uavcan.node.Mode.1.0.uavcan):

```
# Режим работы узла.
uint3 value
uint3 OPERATIONAL = 0
# Нормальный режим.
uint3 INITIALIZATION = 1
```

⁴ Это допустимо, поскольку множества subject-ID и service-ID являются ортогональными, т.е. не влияют друг на друга.

```
# Процесс инициализации; в этом режиме узел находится после включения.
uint3 MAINTENANCE = 2
# Режим обслуживания: калибровка, самодиагностика и др.
uint3 SOFTWARE_UPDATE = 3
# Загрузка новой управляющей программы в узел.
@sealed # означает, что размер типа данных не может быть увеличен
```

Язык DSDL позволяет создавать как простейшие типы данных (применяемые в субмодулях), так и комплексные, включающие в себя множество полей и вложенных типов.

Для модулей и субмодулей мобильного робота были созданы типы данных на языке DSDL с использованием рассмотренных в главе 2 и 3 структур данных для межмодульного взаимодействия. Здесь не рассматриваются служебные типы данных, которые должны публиковаться в составе сервисных сообщений для совместимости модулей с протоколом OpenCyphal, например, сигнал присутствия в сети (`uavcan.node.Heartbeat`).

Сообщение-задание, отправляемое модулем интеллектуального управления (МИУ) на шину, на языке DSDL имеет следующий вид:

Название типа: `modrob.intelligent_module.task.0.1.uavcan`

```
# Сообщение-задание для транспортного модуля
# Идентификатор (subject-ID): 100
float32[2] goal_pos    # [meter] координаты целевой позиции
float32[2] initial_pos  # [meter] координаты исходной позиции
float32 pos_error      # [meter] допустимое отклонение в целевой позиции
uint8 deadline         # [second] предельное время прибытия
@sealed
```

На это сообщение должен подписаться транспортный модуль (по идентификатору), чтобы получать новое задание.

Сообщение-отчёт, отправляемое транспортным модулем на шину, на языке DSDL имеет следующий вид:

Название типа: modrob.transport_module.report.0.1.uavcan

```
# Сообщение-отчёт для модуля интеллектуального управления
# Идентификатор (subject-ID): 105
float32[2] cur_pos      # [meter] координаты текущей позиции
uint8 status           # оценка выполнения задачи
# Константы для статуса
uint8 MOVING_TO_GOAL = 1 # движение к цели
uint8 REACHED_GOAL = 2   # цель достигнута
uint8 EMERGENCY = 3      # аварийный режим
uint8 MALFUNCTION = 4    # неисправность
@sealed
```

На это сообщение должен подписаться МИУ, чтобы получать отчёт о работе транспортного модуля.

Также ТМ должен отправлять на шину сообщение с информацией о своём положении и скорости:

Название типа: modrob.transport_module.pos_and_vel.0.1.uavcan

```
# Сообщение с сенсорными данными транспортного модуля для СМБД
# Идентификатор (subject-ID): 106
float32[2] cur_pos  # [meter] координаты текущей позиции
float32[2] cur_vel  # [meter/sec] текущая скорость ТМ
@sealed
```

На это сообщение может подписаться СМБД для корректировки своих алгоритмов локализации.

Для описания сообщения, отправляемого сенсорным модулем ближнего действия на шину, необходимо ввести вспомогательный тип данных для описания параметров i -го подвижного препятствия на языке DSDL:

Название типа: modrob.sensor_module.obst_info.0.1.uavcan

```
# Параметры подвижного препятствия
float32[2] obst_pos # [meter] координаты центра габаритной окружности
float32[2] obst_vel # [meter/sec] проекции вектора скорости препятствия
```

```
float32 radius      # [meter] радиус габаритной окружности препятствия
@sealed
```

Также необходим вспомогательный тип данных для описания отрезков прямых статических препятствий:

Название типа: modrob.sensor_module.segment_params.0.1.uavcan

```
# Параметры отрезка прямой статического препятствия
float32 x1          # [meter] координата x первого конца отрезка
float32 y1          # [meter] координата y первого конца отрезка
float32 x2          # [meter] координата x второго конца отрезка
float32 y2          # [meter] координата y второго конца отрезка
@sealed
```

Эти вспомогательные типы данных являются частью основного сообщения с сенсорными данными, которое отправляет сенсорный модуль ближнего действия на шину:

Название типа: modrob.sensor_module.sensor_data.0.2.uavcan

```
# Сенсорные данные для ТМ и МИУ
# Идентификатор (subject-ID): 150
uint8 MAX_OBST_NUM = 10      # максимальное количество подвижных препятствий
uint8 MAX_SEG_NUM = 150     # максимальное количество отрезков прямых
float32[2] cur_pos          # [meter] координаты текущей позиции робота
# данные о подвижных препятствиях
modrob.sensor_module.obst_info.0.1[<=MAX_OBST_NUM] obstacles
# данные об отрезках прямых, описывающих неподвижные препятствия
modrob.sensor_module.segment_params.0.1[<=MAX_SEG_NUM] segments
@sealed
```

На это сообщение должен подписаться транспортный модуль, чтобы планировать движение в ближней зоне. На это сообщение может подписаться МИУ для обновления карты местности.

Рассмотрим типы данных DSDL для *субмодулей транспортного модуля*. Для этого сообщения субмодулей, рассмотренные в разделе 3.6, необходимо записать в соответствии с требованиями языка DSDL.

Сообщение когнитивного субмодуля, содержащее в себе заданные скорости и положения для субмодулей-приводов ТМ, имеет следующий вид:

Название типа: modrob.transport_module.cognition_setpoint.0.1.uavcan

```
# Сообщение когнитивного субмодуля для субмодулей-приводов
# Идентификатор (subject-ID): 10
uint8 MAX_ACTUATORS_NUM = 8      # максимальное количество субмодулей-приводов
float32[<=MAX_ACTUATORS_NUM] angular_velocity # [rad/sec] заданные скорости
float32[<=MAX_ACTUATORS_NUM] angular_position # [rad] заданные положения
@sealed
```

На это сообщение должны подписаться все субмодули-приводы, подключенные к шине. Следует отметить, что здесь учитываются только приводы с вращательным движением выходного звена. При этом для линейных приводов эта структура практически не изменится, за исключением названий полей.

Сообщение от субмодуля-привода на языке DSDL имеет вид:

Название типа: modrob.transport_module.feedback.0.1.uavcan

```
# Сообщение субмодуля-привода
# Идентификатор (subject-ID): 15
float32 angular_velocity          # [rad/sec] фактическая скорость
float32 angular_position          # [rad] фактическое положение
uavcan.time.SynchronizedTimestamp timestamp # [usec] метка времени
@sealed
```

Все сообщения от субмодулей-приводов относятся к одной тематической группе с одним идентификатором (subject-ID), используя который когнитивный субмодуль подписывается на сообщения этой группы. При этом подписчик в сети OpenCyphal всегда обладает информацией о том, от какого узла пришло сообщение.

Сообщение от обобщённого сенсорного субмодуля рассмотрим в упрощённом виде, поскольку функционал и структура этого субмодуля требуют отдельного изучения:

Название типа: modrob.transport_module.general_sensor.0.1.uavcan

```
# Сообщение обобщённого сенсорного субмодуля
# Идентификатор (subject-ID): 17
uint8 MAX_SENSORS_NUM = 4           # максимальное количество датчиков
float32[<=MAX_SENSORS_NUM] data    # показания датчиков
@sealed
```

Следует отметить, что согласно спецификации протокола рекомендуется вместо примитивных типов, таких как float32, использовать стандартные типы данных OpenCyphal с явным указанием размерности в названии типа (в соответствии с СИ). В данном случае эта рекомендация не соблюдается, поскольку использование этих типов создаёт дополнительную вложенность структур данных на языке C, что усложняет работу с кодом.

Для конвертации типов данных DSDL в код на конкретном языке программирования (в данном случае C), авторы протокола предоставляют программу Nunavut [40] с открытым исходным кодом. Для всех рассмотренных выше типов данных была выполнена конвертация с помощью Nunavut, и были получены заголовочные файлы (по одному файлу на тип данных), которые можно напрямую подключать в программах для модулей и субмодулей.

Передача рассмотренных сообщений между модулями и субмодулями рассматривается в главе 5.

4.2 Расчёт загруженности коммуникационных каналов

В разделах 3.5 и 3.6 уже проводился расчёт минимальной требуемой пропускной способности коммуникационных сетей (шин) модулей и субмодулей.

Рассчитаем загруженность коммуникационных сетей при учёте служебных данных протокола OpenCyphal и самого стандарта CAN (FD).

К служебным данным протокола OpenCyphal относится информация о передаваемом кадре (номер кадра, начало или конец передачи) и контрольная сумма передачи. Также, если в сообщении есть массивы переменной длины, то в полезной нагрузке непосредственно перед элементами массива всегда размещается количество элементов массива.

Сначала рассчитаем загруженность для шины CAN FD, к которой подключены МИУ, ТМ и СМБД. Силовой модуль в расчёт не принимается, поскольку он публикует на шину только низкочастотные сообщения. Рассмотрим случай, при котором сообщение от СМБД всегда включает в себя данные о 10 подвижных препятствиях, а количество отрезков прямых статических препятствий может различаться и достигать максимального значения (150). В этом случае вместо классической шины CAN необходимо использовать версию CAN FD.

Исходные данные:

- битовая скорость передачи служебных данных (arbitration phase bit rate) – 1 Мбит/с,
- битовая скорость передачи полезных данных (data phase bit rate) – 5 Мбит/с,
- размер полезных данных – 64 байта,
- наибольшая (худшая) длительность одного кадра CAN FD – 186 мкс,
- наименьшая (худшая) частота передачи кадров ($FPS_{\min}$) – 5376 кадров/с.

В таблице 4.2 представлены сообщения модулей, их размеры в байтах, а также число кадров CAN FD, приходящихся на одно сообщение. Сообщение СМБД (sensor_data.0.2.uavcan) имеет различный размер в зависимости от количества отрезков (в скобках после размера сообщения или числа кадров указывается количество отрезков). В таблице приведено сообщение

uavcan.time.Synchronization, которое используется для синхронизации часов МИУ (мастера) с часами остальных модулей.

Таблица 4.2. Характеристики сообщений модулей

Сообщение	Размер сообщения, байт	Число кадров CAN FD
modrob.intelligent_module.task.0.1.uavcan	21	1
modrob.transport_module.report.0.1.uavcan	9	1
modrob.transport_module.pos_and_vel.0.1.uavcan	16	1
modrob.sensor_module.sensor_data.0.2.uavcan	2610 (150), 1810 (100), 1010 (50)	42 (150), 29 (100), 17 (50)
7509.Heartbeat.1.0.uavcan	7	1
7168.Synchronization.1.0.uavcan	7	1

Загруженность шины (BU – bus utilization) модульного робота определяется по следующей формуле [38]:

$$BU = \frac{f_{task} (F_{task} + F_{report}) + f_{sensor} (F_{sensor} + F_{pv}) + nF_h + F_{ts}}{FPS_{min}} \cdot 100\%,$$

где f_{task} – частота публикации сообщения-задания от МИУ (2 Гц),

F_{task} – число кадров, приходящихся на одно сообщение-задание от МИУ,

F_{report} – число кадров, приходящихся на одно сообщение-отчёт от ТМ,

f_{sensor} – частота публикации сообщения СМБД (а также сообщения pos_and_vel от ТМ), Гц,

F_{sensor} – число кадров, приходящихся на одно сообщение СМБД,

F_{pv} – число кадров, приходящихся на одно сообщение с информацией о скорости и положении ТМ,

F_h – число кадров, приходящихся на сообщение Heartbeat,

F_{ts} – число кадров, приходящихся на сообщение синхронизации часов,

n – количество модулей на шине ($n = 4$).

На рисунке 4.2 показаны графики зависимости загруженности шины CAN FD от частоты публикации сообщений СМБД (f_{sensor}) для трёх случаев: при

150, 100 и 50 отрезках прямых статических препятствий, информация о которых содержится в одном сообщении СМБД.

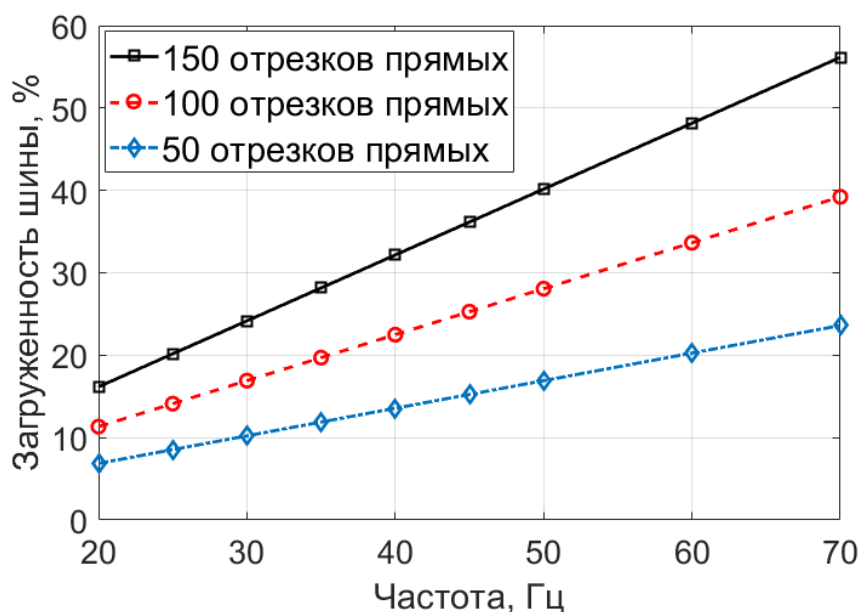


Рисунок 4.2. Загруженность шины полнофункциональных модулей при использовании OpenCyphal

Видно, как влияет на загруженность шины число отрезков прямых, данные о которых содержатся в сообщении от СМБД: при 150 отрезках и частоте 50 Гц шина CAN FD загружена на 40.2%, при 50 – загруженность меньше 17%.

Для рассматриваемой рабочей частоты отправки сенсорных данных $f_{\text{sensor}} = 20$ Гц загруженность шины не превышает 17% при максимальном размере сообщения от СМБД. Оставшийся запас пропускной способности (около 80%) необходим для других модулей робота (например, для модуля-манипулятора) и передачи различных сервисных сообщений. Например, сообщений, содержащих информацию о конфигурации модулей, калибровочные данные и т.д.

Длительность передачи одного сообщения сенсорного модуля ближнего действия при его максимальном размере (42 кадра) составляет 7.8 мс. Соответственно транспортный модуль будет получать новые сенсорные данные с задержкой порядка 8 мс.

Рассчитаем загруженность шины CAN, к которой подключены *субмодули транспортного модуля*.

Как и в разделе 3.6 будем считать, что к шине подключен один когнитивный субмодуль, один обобщённый сенсорный субмодуль и от 2 до 8 субмодулей-приводов (по количеству колёс ТМ).

В таблице 4.3 представлены сообщения субмодулей, их размеры в байтах, а также число кадров CAN, приходящихся на одно сообщение. Сообщение когнитивного субмодуля (setpoint.0.1.uavcan) имеет различный размер в зависимости от числа субмодулей-приводов (в скобках после размера сообщения или числа кадров указывается количество субмодулей).

Таблица 4.3. Характеристики сообщений субмодулей

Сообщение	Размер сообщения, байт	Число кадров CAN
modrob.transport_module.setpoint.0.1.uavcan	18(2), 26(3), 34(4), 50(6), 66(8)	3(2), 4(3), 6(4), 8(6), 10(8)
modrob.transport_module.general_sensor.0.1.uavcan	17	3
modrob.transport_module.feedback.0.1.uavcan	15	3
7509.Heartbeat.1.0.uavcan	7	1
7168.Synchronization.1.0.uavcan	7	1

Загруженность шины транспортного модуля определяется по следующей формуле [39]:

$$BU_{TM} = \frac{\left[f(F_k(n) + nF_{np} + F_c) + (2 + n)F_h + F_{ts} \right] \cdot l_{frame}}{V_{max}} \cdot 100\%,$$

где f – частота публикации сообщений setpoint.0.1.uavcan,

general_sensor.0.1.uavcan и feedback.0.1.uavcan на шину, Гц,

V_{max} – битовая скорость передачи данных (1000000 бит/с),

n – количество субмодулей-приводов на шине,

$F_k(n)$ – число кадров (frames), приходящихся на одно сообщение

setpoint.0.1.uavcan (зависит от n),

F_{np} – число кадров, приходящихся на одно сообщение feedback.0.1.uavcan,

F_c – число кадров, приходящихся на одно сообщение general_sensor.0.1.uavcan,

$l_{frame} = 144$ – длина одного кадра CAN в битах в худшем случае (с учётом битовых подстановок для кодировки NRZ).

На рисунке 4.3 показаны графики зависимости загруженности классической шины CAN от частоты публикации f субмодулей в диапазоне от 10 до 160 Гц и от количества субмодулей-приводов (2, 3, 4, 6 и 8).

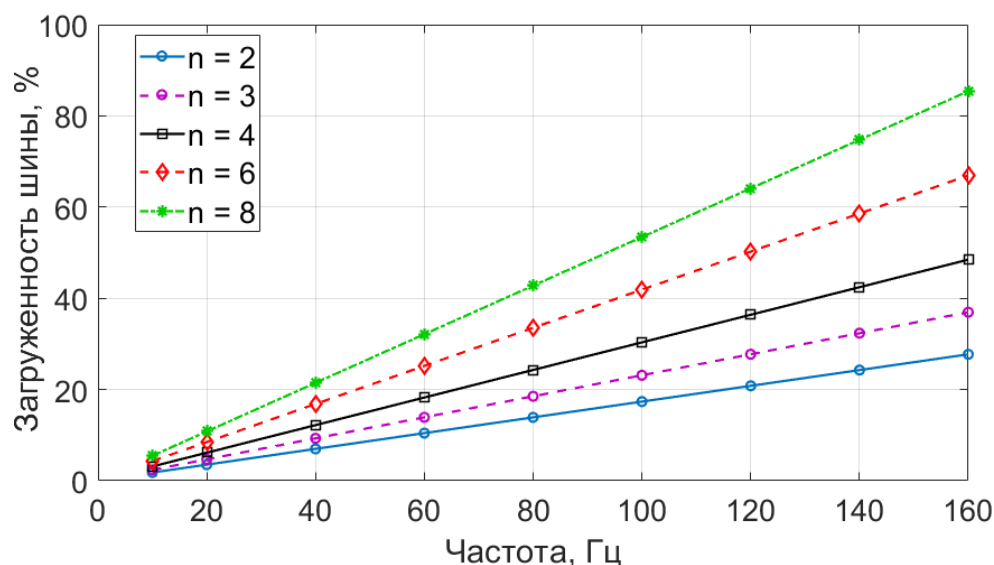


Рисунок 4.3. Загруженность шины субмодулей при использовании протокола OpenCyphal

Из графиков видно, что загруженность шины достигает 80% для случая 8 субмодулей-приводов на шине уже при частоте публикации 150 Гц. При этом без учёта служебных данных протокола такая загруженность возникала только при частоте 500 Гц (см. рисунок 3.17 в разделе 3.6). Следует отметить, что частота публикации сообщений на шину выше 100 Гц является скорее исключением и нужна для достаточно быстродействующих систем. Например, при максимальной скорости робота 2 м/с, позиционная ошибка будет составлять всего лишь порядка 0.02 м при задержке 0.01 с.

Для рассматриваемой конфигурации ТМ нас интересует случай при $n = 4$: загруженность шины на диапазоне частот до 160 Гц не превышает 50%. Это означает, что кроме основных сообщений, связанных с непосредственным управлением приводами, на шину допускается публиковать больше сервисных сообщений благодаря запасу пропускной способности. Например, если

субмодуль-привод фиксирует резкое превышение допустимой температуры двигателя, он может начать публиковать сообщение, предназначенное для когнитивного субмодуля, с информацией о текущей температуре, токе и напряжении. При этом согласно принципу полной функциональности эта информация не должна отправляться когнитивному субмодулю в штатном режиме работы.

4.3 Подключение модулей в режиме «plug and play»

Для подключения модулей в режиме «plug and play» интерфейсы модулей должны удовлетворять целому ряду требований, связанных с электромеханической совместимостью разъёмов и соединителей. В данном разделе рассмотрим только программные аспекты рассматриваемой проблемы.

Протокол OpenCyphal позволяет подключать модули в систему в режиме «plug and play». Необходимым условием является совместимость модуля с протоколом. Согласно спецификации каждый модуль должен каждую секунду отправлять в сеть сигнал присутствия (сообщение `uavcan.node.Heartbeat`), что является минимальным требованием к узлу.

Алгоритм подключения модулей к сети OpenCyphal подробно описывается в спецификации протокола. Рассмотрим ключевые особенности этого алгоритма.

Модуль, подключаемый в режиме «plug and play», не имеет заранее установленного идентификатора в сети (node-ID). Однако каждый модуль должен иметь уникальный идентификатор устройства (unique-ID)⁵ с длиной 128 бит.

Для получения идентификатора в сети после физического подключения модуля к шине в OpenCyphal используется специальный протокол, опирающийся на так называемые *анонимные сообщения*. В анонимных сообщениях не указывается идентификатор узла-отправителя. При этом в сети должен быть как минимум один модуль-распределитель с фиксированным node-ID, который будет

⁵ Предполагается, что среди всех совместимых с OpenCyphal устройств вероятность наличия двух модулей с одинаковыми идентификаторами пренебрежимо мала.

осуществлять процесс подключения остальных модулей в режиме «plug and play». Этот модуль должен вести таблицу всех подключенных устройств с их идентификаторами unique-ID и node-ID. Распределение идентификаторов осуществляется с помощью служебных сообщений группы uavcan.pnp. При этом для обеспечения большей надёжности в сети может быть несколько модулей-распределителей, которые используют копии таблицы с идентификаторами.

В рассматриваемой в работе модульной архитектуре модулем-распределителем на втором уровне иерархии (см. рисунок 3.16) должен быть модуль интеллектуального управления. При этом когнитивный субмодуль транспортного модуля в сети на втором уровне может динамически получать идентификатор в соответствии с режимом «plug and play». Однако внутри своей сети на первом уровне когнитивный субмодуль должен иметь *фиксированный* node-ID, поскольку на этом уровне он уже сам является распределителем идентификаторов для остальных субмодулей. Это может быть заранее установленный идентификатор, который специально оговаривается в спецификации к драйверу.

4.4 Выводы по четвёртой главе

В главе был рассмотрен протокол ModRob, который разрабатывался автором совместно с коллегами. К достоинствам протокола можно отнести простоту реализации (поддерживаются только однокадровые сообщения) и небольшой набор команд (5 команд). Протокол позволял субмодулям работать в режиме реального времени (эксперимент подробно рассматривается в последней главе), однако имел определённые недостатки. Кроме того, протокол не мог конкурировать с такими широко известными протоколами как CANopen или OpenCyphal (UAVCAN). Тем не менее, сам процесс разработки протокола ModRob позволил определить необходимые и достаточные требования к протоколу межмодульной коммуникации в предложенной архитектуре СУ

модульного МР. В связи с этим было решено перейти к готовому протоколу OpenCyphal, который соответствует всем сформулированным требованиям по межмодульной коммуникации и обладает рядом преимуществ по сравнению с CANopen, SAE J1939 и CANaerospace.

Для обмена сообщениями в сети OpenCyphal используются специальные типы данных на языке DSDL. Для модулей и субмодулей мобильного робота были созданы типы данных, совместимые с OpenCyphal, на основе рассмотренных в главе 2 и 3 структур данных для межмодульного взаимодействия МИУ, ТМ и СМБД.

Для каждого разработанного типа данных был определён размер сообщения в кадрах CAN в соответствии со спецификацией протокола. Был проведён уточнённый расчёт загруженности шины CAN FD на уровне полнофункциональных модулей, а также шины CAN на уровне субмодулей ТМ. Расчёт показал, что в диапазоне частот передачи сенсорных данных от 20 до 70 Гц загруженность шины CAN FD не превышает 60% при максимальном размере сообщения, которое публикуется сенсорным модулем ближнего действия. На уровне субмодулей загруженность шины CAN достигает 80% при наличии 10 субмодулей, которые публикуют сообщения с частотой 150 Гц. Для рассматриваемой в диссертации конфигурации транспортного модуля с четырьмя меканум-колёсами загруженность шины не превышает 50% при частоте публикации 150 Гц.

Показано, что для подключения модулей/субмодулей в режиме «plug and play» следует использовать готовый набор служебных сообщений протокола OpenCyphal. В спецификации протокола подробно рассмотрен метод распределения уникальных идентификаторов между модулями в сети. При этом есть ряд ограничений, например, в сети OpenCyphal должен быть хотя бы один узел с фиксированным идентификатором.

ГЛАВА 5. МОДЕЛИРОВАНИЕ И ЭКСПЕРИМЕНТАЛЬНЫЕ ИССЛЕДОВАНИЯ

В главе рассматривается разработанная компьютерная модель системы управления модульного мобильного робота [56], в основу которой положены модели конечных автоматов полнофункциональных модулей, рассмотренные в главе 3. Диаграммы состояний автоматов для МИУ и ТМ реализуются в блоках Simulink без учёта некоторых вспомогательных состояний. Приведены результаты компьютерных симуляций движения мобильного робота с модульной архитектурой при различных условиях: перемещение в среде с подвижными и неподвижными препятствиями, перемещение в среде с тупиком, симуляция столкновения робота с препятствием.

В главе приведены описания и характеристики разработанных лабораторных макетов и прототипов модулей и субмодулей. Часть изготовленных устройств используется в экспериментах, которые описываются в конце главы.

Приводится результат эксперимента с разработанным протоколом ModRob, краткое описание которого было представлено в разделе 4.1. Протокол позволяет субмодулям ТМ функционировать в реальном времени [7]. Тем не менее, в силу ряда причин, описанных в разделе 4.1, было решено использовать протокол OpenCyphal для межмодульного взаимодействия.

В последней части главы описывается эксперимент, цель выполнения которого заключалась в определении интервала времени с момента передачи сенсорных данных от СМБД до момента приёма сообщения от когнитивного субмодуля одним из базовых субмодулей для оценки соответствия этого интервала времени параметрам компьютерной модели. В управляющих программах модулей и субмодулей реализован протокол OpenCyphal и используются типы данных сообщений, которые были также рассмотрены в разделе 4.1.

5.1 Компьютерная модель системы управления модульного робота

Для оценки работоспособности предложенной модульной архитектуры было проведено компьютерное моделирование в программной среде MATLAB (Simulink). Для моделирования работы диаграмм состояний UML используется библиотека Stateflow.

Проводилась симуляция движения робота, состоящего из транспортного модуля (ТМ), модуля интеллектуального управления (МИУ) и сенсорного модуля ближнего действия (СМБД), в плоской среде с подвижными и неподвижными препятствиями. Структурная схема модели показана на рисунке 5.1.

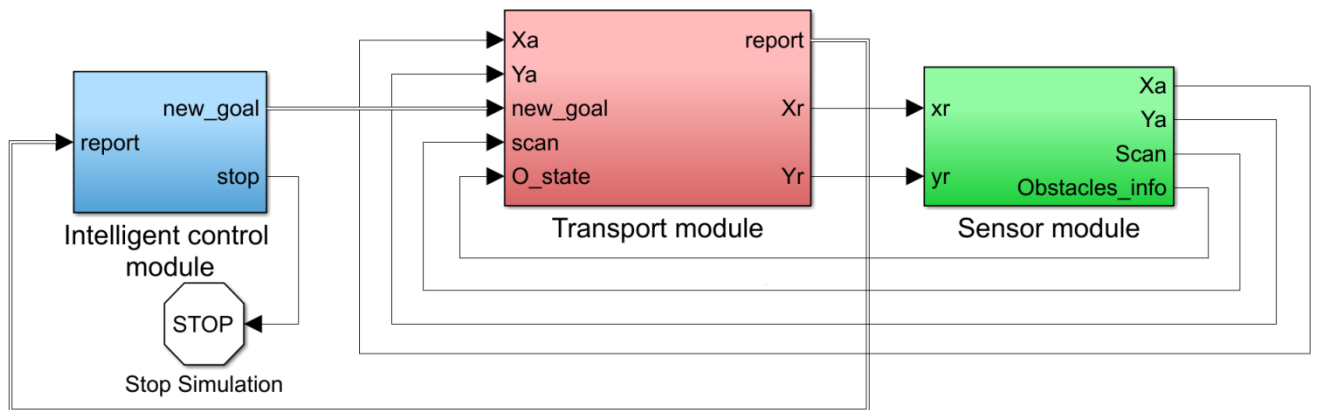


Рисунок 5.1. Компьютерная модель мобильного робота, состоящего из трёх основных модулей

Основные блоки модели:

1. *Intelligent control module* – блок, выполняющий часть функций МИУ: хранит список точек маршрута робота, сравнивает положение робота, получаемое от ТМ, с заданным. Сообщение-отчёт (report) транспортного модуля является входным сигналом для блока и представляет собой вектор-строку:

$$\text{report} = [\text{state } X_a \ Y_a],$$

где state – состояние конечного автомата ТМ,

X_a , Y_a – координаты текущего положения робота (ТМ).

В компьютерной модели принимается, что ТМ получает информацию о своём положении с помощью одометрии. Если очередная целевая позиция

достигнута с допустимой погрешностью (не более 5 см), МИУ отправляет в ТМ координаты следующей точки пути (выход с названием «MSG»).

В основе блока «Intelligent control module» лежит диаграмма состояний автомата МИУ (рисунок 5.2), отражающая лишь часть функций диаграммы, представленной в разделе 3.1. В модели МИУ не строится новый маршрут на карте, вместо этого используются заранее подготовленные наборы точек, принадлежащие тому или иному пути.

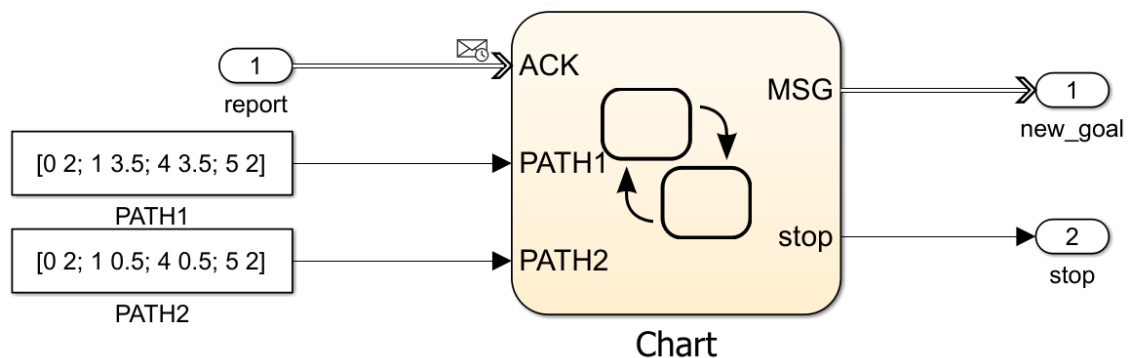


Рисунок 5.2. Блок «Intelligent control module»

Когда робот достигает последней позиции, на выходе «stop» устанавливается положительное значение, после чего компьютерная симуляция завершается.

В диаграмме состояний (рисунок 5.3) не учитывается «Аварийный останов», и реализованы только три состояния рабочего режима: «Idle» («Ожидание задачи»), «Planning» («Планирование»), «Control» («Контроль») и «Task_complete» («Задача выполнена»).

В модели переход от ожидания к планированию происходит через 1 секунду (принимается, что оператор отправляет задание через такой интервал). В состоянии «Planning» происходит простое переключение между двумя путями, например, если один из путей заблокирован или произошло столкновение. Переход к состоянию «Control» происходит через две секунды после смены пути (принимается, что на планирование маршрута уходит 2 секунды). Путь робота задаётся как матрица следующего вида:

$$PATH = \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ \dots & \dots \\ x_n & y_n \end{bmatrix},$$

где x_j, y_j – координаты j -й точки пути, $j = 1, 2, \dots, n$.

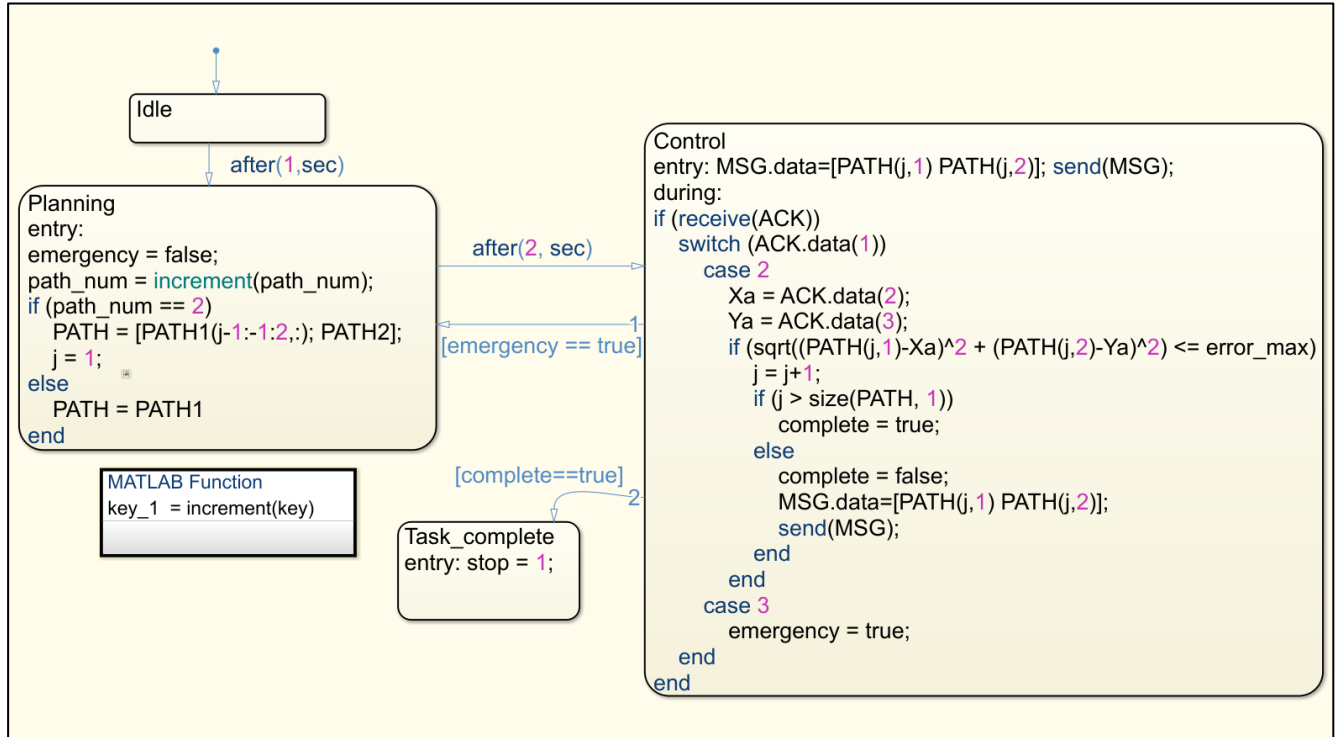


Рисунок 5.3. Модель конечного автомата МИУ, построенная с помощью библиотеки Stateflow

Основная работа блока МИУ происходит в состоянии «Control». При входе в состояние автомат сразу отправляет первую точку пути. Автомат ждёт очередное сообщение-отчёт от ТМ и выполняет алгоритм, блок-схема которого представлена на рисунке 5.4. В алгоритме не учтён счётчик ошибок ТМ, поскольку в модели сделано допущение, что ТМ всегда достигает цели с приемлемой точностью. В состоянии «Control» МИУ после получения очередного отчёта от ТМ вычисляет позиционную ошибку прибытия в цель, поскольку предполагается, что МИУ обладает более точной информацией о среде и реальном положении робота. Если очередная цель достигнута с допустимой ошибкой, модуль интеллектуального управления отправляет новую цель в виде

сообщения (MSG) с помощью специальной функции «send», которая используется в диаграммах состояний UML библиотеки Stateflow.

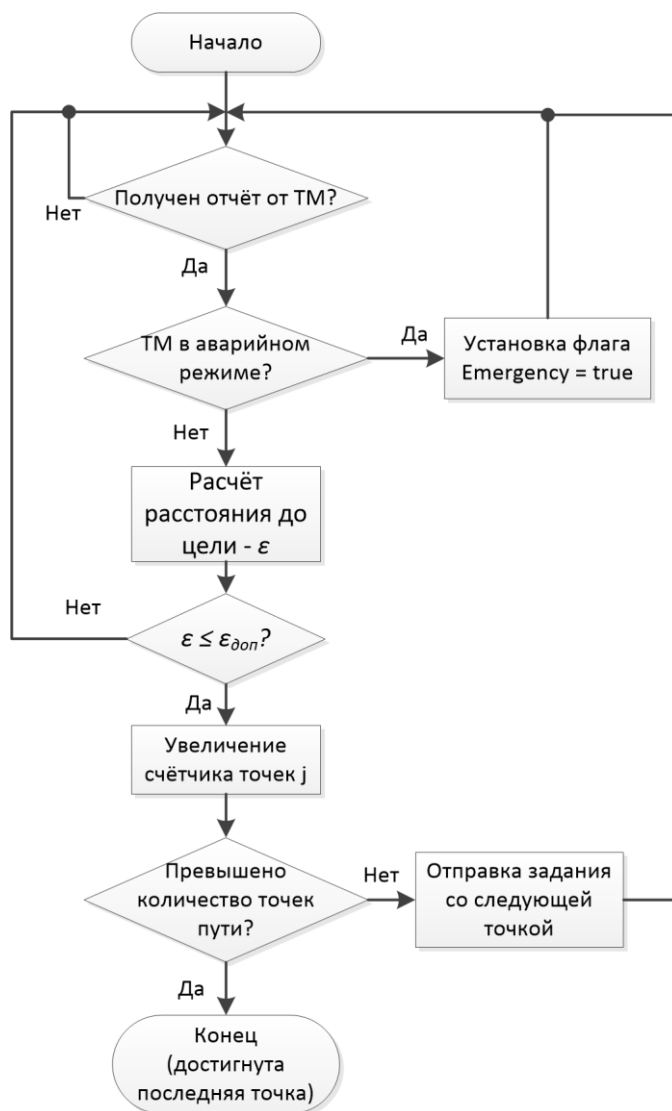


Рисунок 5.4. Блок-схема алгоритма, выполняемого в состоянии «Control» автомата МИУ

В блок-схеме ε – Евклидово расстояние от текущего положения ТМ с координатами X_r , Y_r до целевой точки с координатами X_i , Y_i , $\varepsilon_{доп}$ – величина допустимой ошибки по положению в целевой точке (здесь равна 5 см).

Как видно из блок-схемы и диаграммы состояний переход в состояние «Planning» происходит, если устанавливается флаг экстренного режима (emergency). Флаг устанавливается, если ТМ сообщает об аварийном режиме.

2. *Transport module* – блок, реализующий основные функции транспортного модуля: выполнение алгоритма объезда препятствий, рассмотренного в параграфе

3.4, и закон движения, заданный по формуле (3.3). Блок принимает на вход сообщение от блока МИУ с координатами целевой позиции (new_goal): X_g , Y_g ; координаты текущего положения робота – X_a , Y_a ; данные о подвижных препятствиях, представленные в виде матрицы «O_state» размерностью $m \times 5$, где m – количество препятствий в ближней зоне ТМ, а также параметры отрезков прямых («scan»), которые описывают неподвижные препятствия вокруг робота в пределах ближней зоны СМБД.

Блок «Transport module» состоит из следующих основных компонентов (рисунок 5.5): модели конечного автомата ТМ («Chart»), планировщика движения («TM local planner») и блока с динамической моделью колёсной платформы ТМ («TM omniwheel platform»).

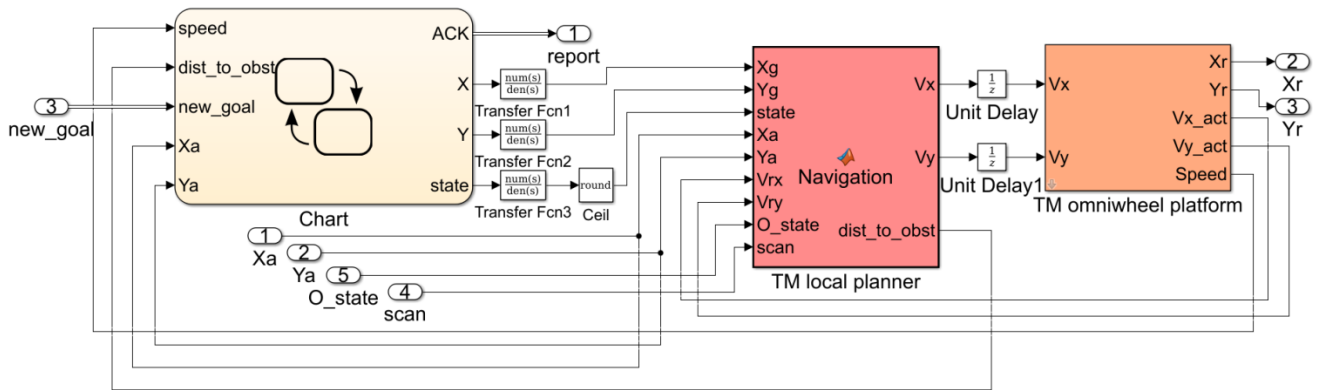


Рисунок 5.5. Блок «Transport module»

Модель конечного автомата ТМ (рисунок 5.6) включает в себя только подсостояния рабочего режима и «Аварийный режим» («Emergency»). При этом подсостояния «Idle», «Moving_to_goal» и «Reached_goal» соответствуют подсостояниям диаграммы ТМ на рисунке 3.4: «Ожидание цели», «Перемещение к цели» и «Цель достигнута».

При входе в каждое состояние и подсостояние изменяется переменная, характеризующая состояние (state):

- state = 0 – Idle ($S_{T.1.1.0}$),
- state = 1 – Moving_to_goal ($S_{T.1.1.1}$),
- state = 2 – Reached_goal ($S_{T.1.1.2}$),
- state = 3 – Emergency (Stop) ($S_{T.1.2}$),

- state = 4 – Back_away (отъезд назад после столкновения).

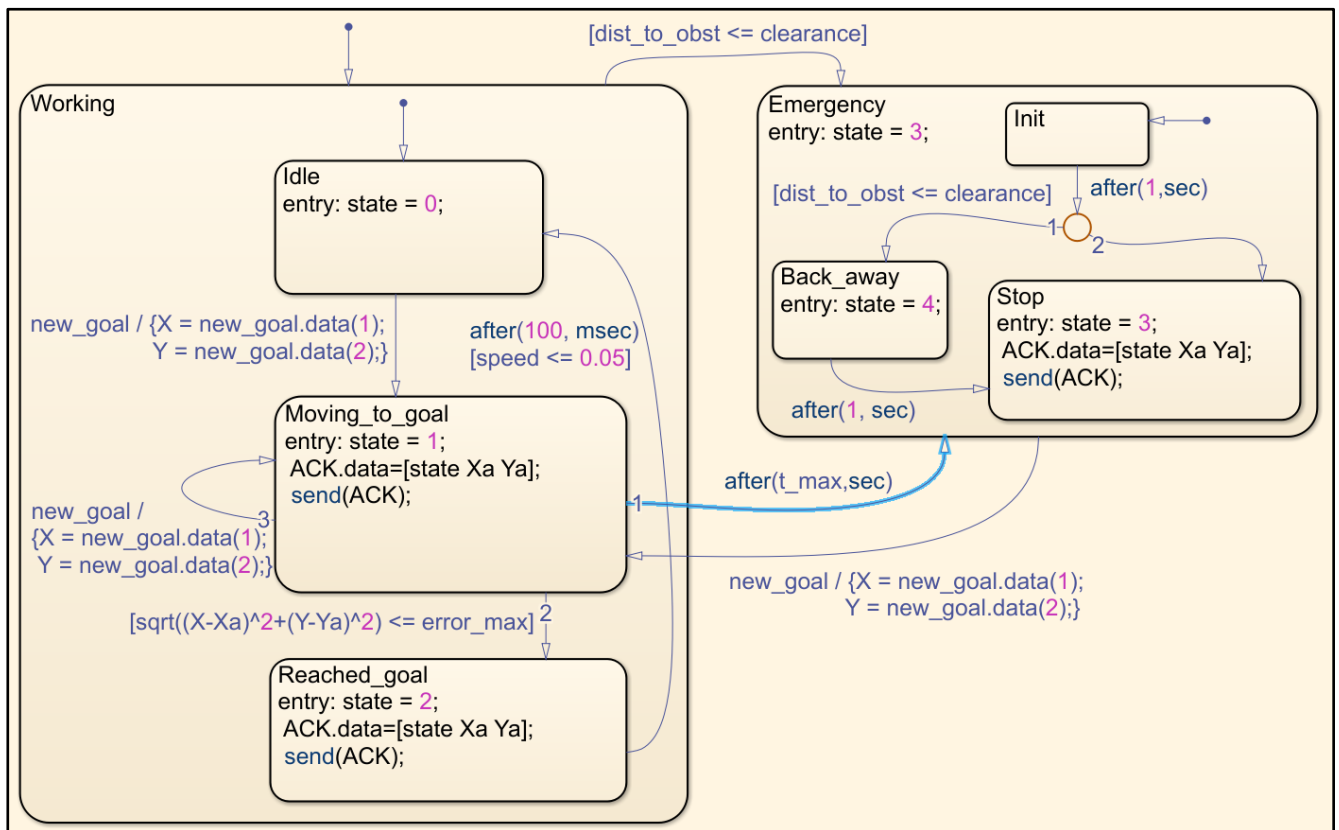


Рисунок 5.6. Модель конечного автомата ТМ

В данной модели «Аварийный режим» является составным состоянием. При входе в него автомат оказывается в подсостоянии «Init», затем спустя 1 секунду происходит переход в подсостояние «Back_away» или в конечное подсостояние «Stop» (если столкновения не было). В подсостоянии «Back_away» в течение 1 секунды ТМ отъезжает в противоположную сторону от столкновения. После этого происходит переход в подсостояние «Stop» и отправка отчёта в МИУ.

Модель конечного автомата неразрывно связана с блоками «ТМ local planner» и «ТМ omniwheel platform», от которых она получает данные о фактическом положении ТМ, скорости и расстоянии до ближайшего подвижного препятствия (dist_to_obst). Соответственно если это расстояние меньше минимального допустимого расстояния (clearance), то автомат фиксирует столкновение. Модель конечного автомата выдаёт для блока ТМ local planner координаты целевой точки и состояние автомата (state).

Блок «*TM local planner*» использует информацию о состоянии конечного автомата для выбора закона управления движением модуля. Для состояния «Moving_to_goal» ($state = 1$) скорость платформы рассчитывается согласно алгоритму 3.1 (см. раздел 3.4) по формуле (3.7). Область достижимых скоростей RV (см. рисунок 3.12) в блоке представлена в виде матрицы размером 100 на 2, которая получена в результате прямого (декартова) произведения двух векторов длиной 10 элементов.

Для состояний «Idle», «Reached_goal» и «Emergency» скорость платформы должна быть равна нулю:

$$\mathbf{v}_r^{new} = (0, 0)^T.$$

Для подсостояния отъезда назад (Back_away) скорость платформы определяется выражением:

$$\mathbf{v}_r^{new} = \begin{pmatrix} -\text{sgn}(d_{colX}) \cdot v_{back} \cdot \frac{|d_{colX}|}{\max(|d_{colX}|, |d_{colY}|)} \\ -\text{sgn}(d_{colY}) \cdot v_{back} \cdot \frac{|d_{colY}|}{\max(|d_{colX}|, |d_{colY}|)} \end{pmatrix},$$

где $\text{sgn}(x)$ – функция, возвращающая знак x ,

v_{back} – максимальная скорость отъезда,

$d_{colX} = X_{ob} - X_r$ и $d_{colY} = Y_{ob} - Y_r$ – расстояния по осям базовой системы координат между центрами робота и препятствия, с которым произошло столкновение, где X_{ob} , Y_{ob} – координаты центра подвижного препятствия, X_r , Y_r – координаты текущего положения ТМ.

Проекции линейной скорости передаются на выход блока «*TM local planner*».

3. *TM omniwheel platform* – блок, содержащий динамическую модель ТМ с всенаправленным приводом. Блок получает на вход проекции заданной скорости и выдаёт фактические скорости, а также координаты положения платформы.

4. *Sensor module* – блок, выполняющий часть функций сенсорного модуля ближнего действия (СМБД) и включающий в себя модель лазерного сканера

«Sensor_Scanner», модели подвижных препятствий («Obstacle1», «Obstacle2», «Obstacle3») и блок визуализации движения робота в среде («Visualizer»).

Блок «Sensor_Scanner» (рисунок 5.7) сканирует пространство вокруг робота с помощью идеальных дальномеров. Блок возвращает матрицу с параметрами отрезков прямых, относящихся к статическим препятствиям:

$$scan = \begin{bmatrix} x_{11} & y_{11} & x_{12} & y_{12} \\ x_{21} & y_{21} & x_{22} & y_{22} \\ \dots & \dots & \dots & \dots \\ x_{n1} & y_{n1} & x_{n2} & y_{n2} \end{bmatrix},$$

где x_{i1}, y_{i1} – координаты первого конца i -го отрезка,

x_{i2}, y_{i2} – координаты второго конца i -го отрезка,

n – количество отрезков прямых.

Каждый блок подвижного препятствия выдаёт информацию о состоянии препятствия («Obstacle_State») в соответствии с таблицей 2.4 (раздел 2.4). Данные каждого препятствия группируются в матрицу («Obstacles_info»).

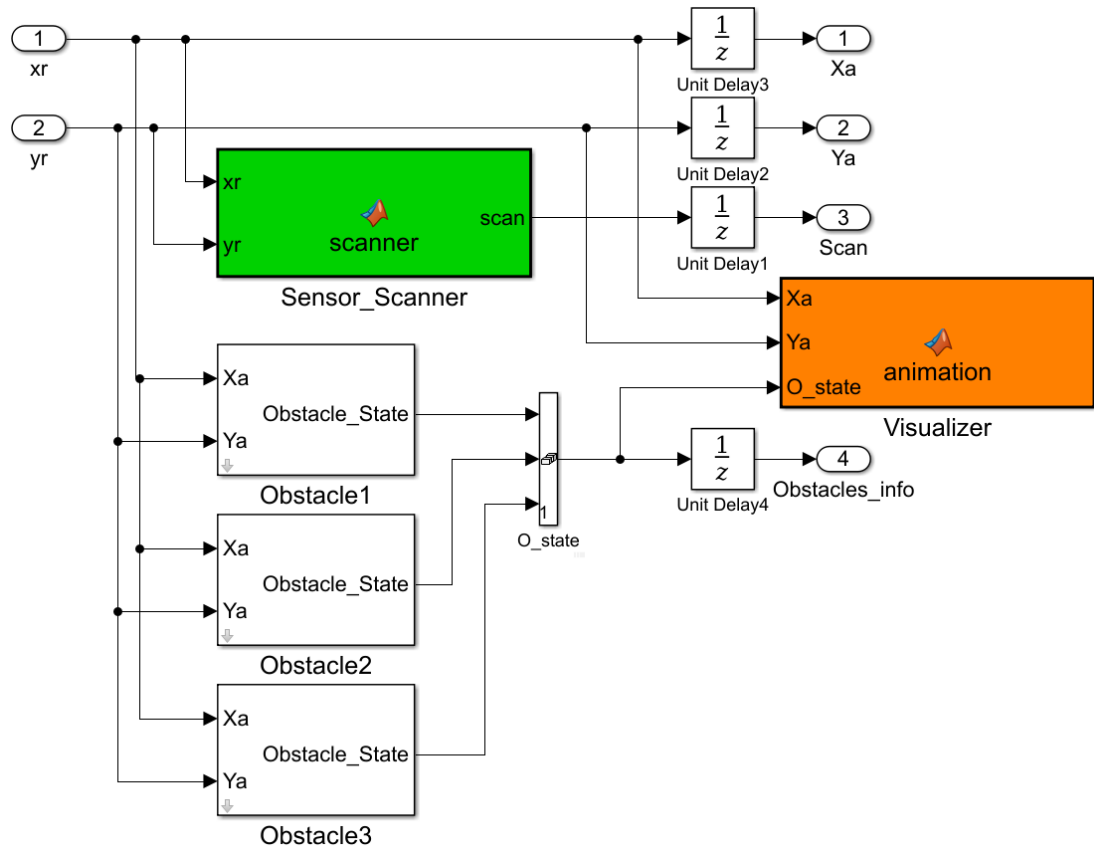


Рисунок 5.7. Блок Sensor module

Координаты робота используются для блоков подвижных препятствий, чтобы контролировать их поведение во время симуляций (например, остановка препятствия при столкновении с роботом).

Блоки «Unit Delay» используются для имитации задержки передачи данных от СМБД к ТМ.

5.2 Исследование компьютерной модели

Исходные параметры модели:

1. Максимальная линейная скорость робота – 0.5 м/с.
2. Максимальное линейное ускорение робота – 4.65 м/с^2 .
3. Радиус габаритной окружности робота – 0.15 м.
4. Задержка выдачи задающих воздействий на приводы – 0.01 с.
5. Задержка передачи данных от СМБД в ТМ – 0.05 с.
6. Минимальное допустимое расстояние до препятствия (clearance) – 0.05 м.
7. Максимальное число достижимых скоростей (RV) – 100.

Количество отрезков прямых n , которое выдаёт блок «Sensor_Scanner» СМБД, зависит от модели среды и в проводимых симуляциях не превышало 20-30 отрезков.

Сначала рассмотрим работу модели при движении модульного мобильного робота в среде, в которой есть только движущиеся препятствия (статических препятствий нет).

Пусть ТМ должен переместиться из точки с координатами (0, 0) в точку с координатами (3, 0). Целевая точка задаётся в блоке МИУ – «Intelligent control module». Модель СМБД отправляет данные о трёх препятствиях, движущихся в ближней зоне ТМ. Здесь и далее делаем допущение о том, что подвижные препятствия не реагируют на робот и движутся по прямой. Расположение препятствий и направления их скоростей показаны на рисунке 5.8.

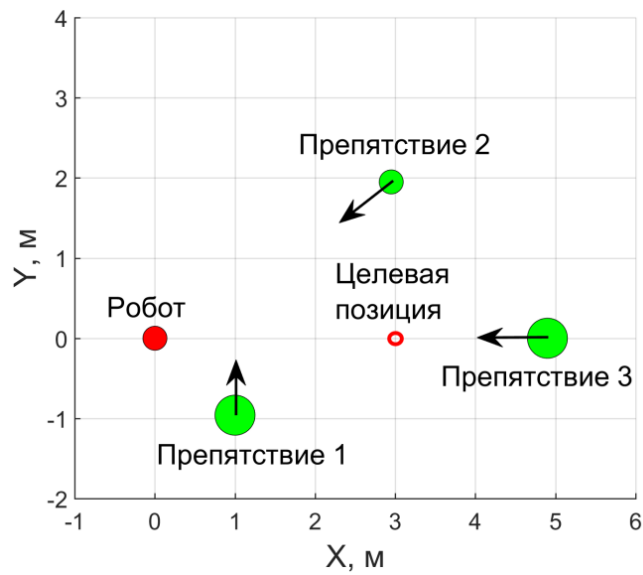


Рисунок 5.8. Расположение препятствий и робота

Скорости препятствий не превышают максимальной линейной скорости робота. Траектория модульного робота и препятствий после проведения компьютерной симуляции показана на рисунке 5.9. Все препятствия начали движение одновременно с роботом.

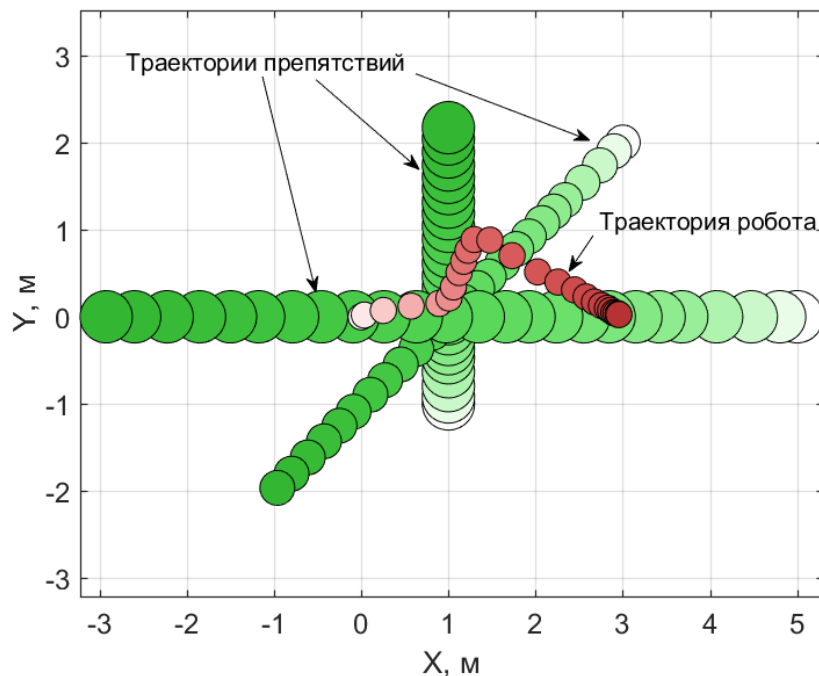


Рисунок 5.9. Траектория движения робота при объезде движущихся препятствий; робот и препятствия обозначаются цветными кругами, которые имеют более яркие цвета в начальный момент времени, тёмные цвета – в конечный момент

Видно, что модульный робот успешно преодолел все препятствия и достиг целевой позиции с координатами (3, 0).

Рассмотрим ещё один случай движения, когда второе препятствие находится в точке (2, 1) и начинает движение в отрицательном направлении оси Y только через 2 секунды с момента запуска робота. Чтобы избежать столкновения, роботу пришлось остановиться и выждать некоторое время, прежде чем продолжить перемещение к цели.

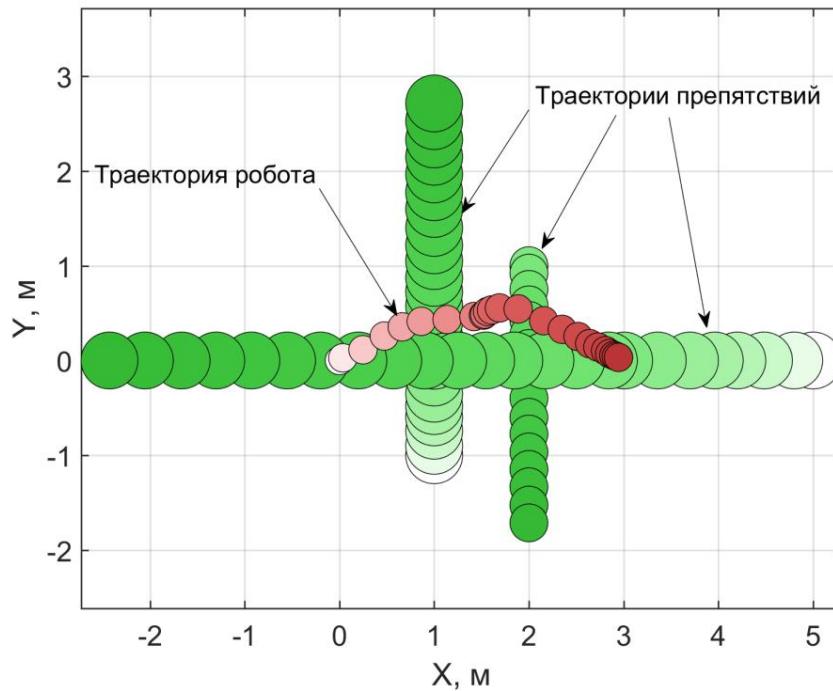


Рисунок 5.10. Траектория движения робота при объезде движущихся препятствий

Теперь рассмотрим движение робота в среде при наличии как подвижных, так и неподвижных препятствий (рисунок 5.11). В начальный момент робот находится в точке с координатами (-0.5, -0.5). Пусть МИУ спланировал маршрут, состоящий из трёх точек со следующими координатами: (1.5, 2), (4.5, 2) и (6, -0.75). ТМ (робот) должен последовательно пройти эти позиции. ТМ получает от МИУ очередную целевую точку после того, как расстояние между фактическим положением робота и целевым становится меньше 5 см.

В среде присутствует одно препятствие, которое движется влево с позиции с координатами (5, 2). Оно начинает движение через 6 секунд после запуска робота, направляется влево и останавливается в точке (2, 2).

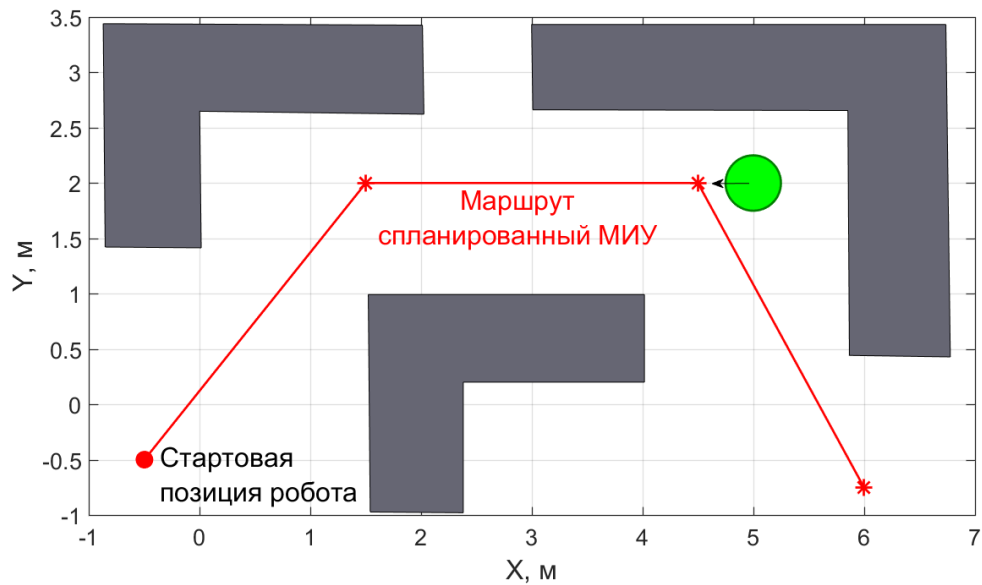


Рисунок 5.11. Конфигурация среды со спланированным маршрутом робота

Траектория модульного робота и препятствия после компьютерной симуляции показана на рисунке 5.12.

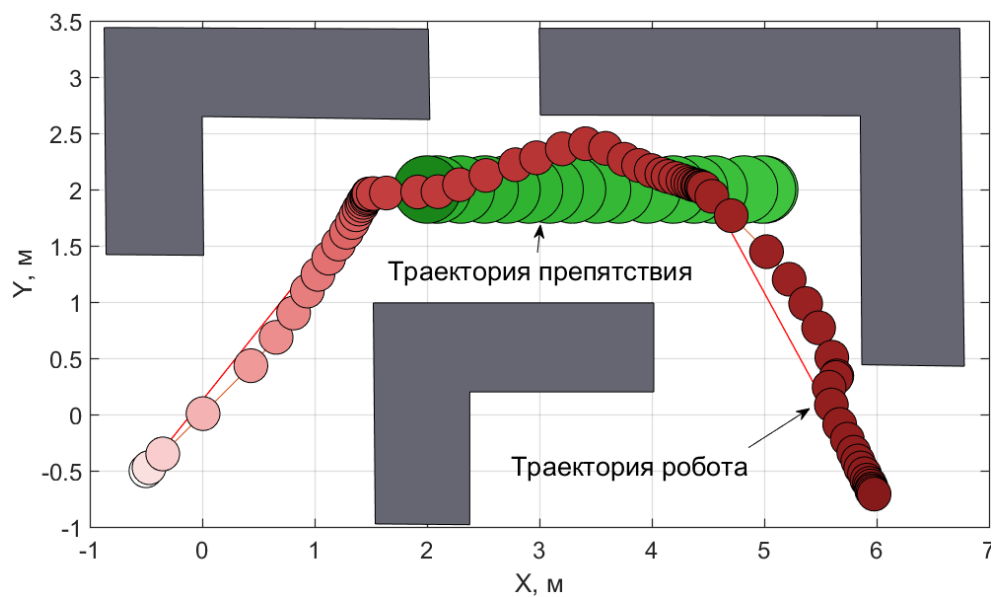


Рисунок 5.12. Движение модульного робота в среде со статическими препятствиями и одним подвижным препятствием

Из графиков на рисунке видно, что робот успешно объехал движущееся препятствие и избежал столкновения со стенками в процессе объездного манёвра.

Рассмотрим следующую ситуацию. Робот находится в помещении с одним неподвижным прямоугольным препятствием в центре (рисунок 5.13).

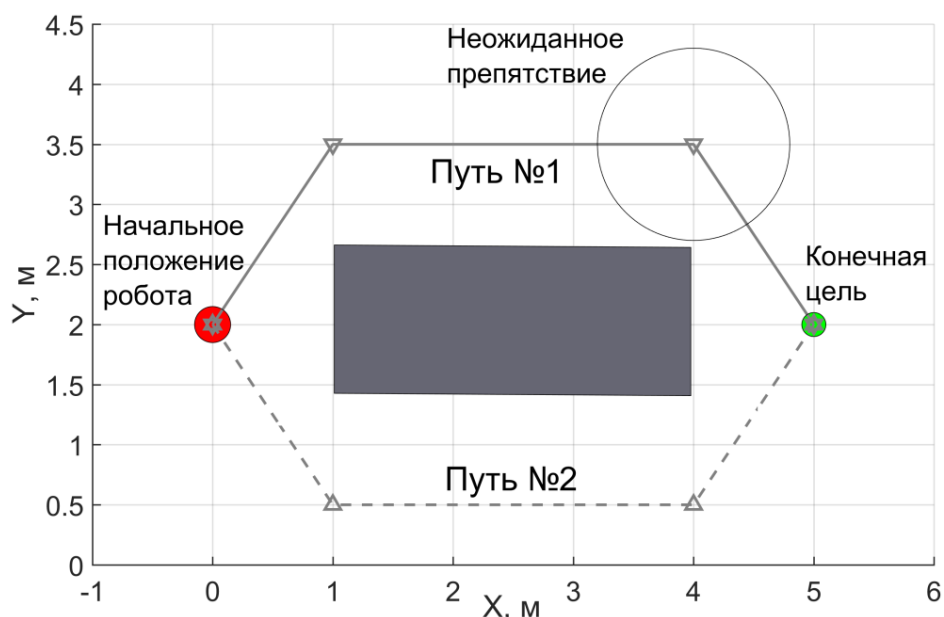


Рисунок 5.13. Модель среды с одним известным статическим препятствием и одним неизвестным подвижным препятствием

В карте местности это препятствие учтено и МИУ спланировал два одинаковых маршрута. Пусть МИУ выбрал путь №1 с координатами целевых точек (1, 3.5), (4, 3.5) и (5, 2). ТМ должен последовательно посетить эти точки. При этом навстречу ему начало движение большое препятствие с позиции (4, 3.5) со скоростью, превышающей максимальную скорость ТМ. В симуляции принимается, что это препятствие не могло быть учтено на этапе построения пути (например, оно появилось после обновления карты).

На рисунке 5.14 показана траектория модульного робота после проведения симуляции.

Сначала ТМ двигался к точке (1, 3.5) в штатном режиме ($\text{state} = 1$), но движущееся навстречу препятствие не позволило добраться до первой точки с координатами (1, 3.5) и произошло столкновение с препятствием. Из-за столкновения конечный автомат транспортного модуля изменил своё состояние на «Emergency» и выполнил останов. Затем ТМ отъехал назад на небольшое расстояние (подсостояние «Back_away»), после чего остановился и отправил отчёт в МИУ.

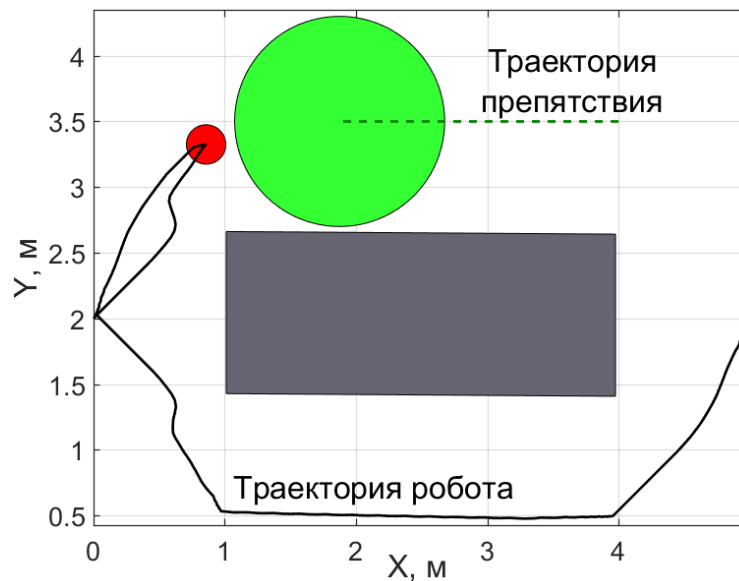


Рисунок 5.14. Траектория робота после проведения симуляции столкновения

МИУ проанализировал отчёт ТМ и зафиксировал экстренную ситуацию (установка флага emergency). Автомат МИУ перешёл из состояния «Control» в состояние «Planning», в котором выбрал второй путь, аналогичный первому. МИУ начал отправлять в ТМ координаты целевых точек второго пути, при этом первая точка, которую должен был посетить ТМ, являлась исходной позицией робота перед началом симуляции (точка (0, 2)). В итоге робот достиг целевой позиции, и автомат МИУ перешёл в состояние «Task_complete».

На рисунке 5.15 показан график изменения состояний автомата ТМ.

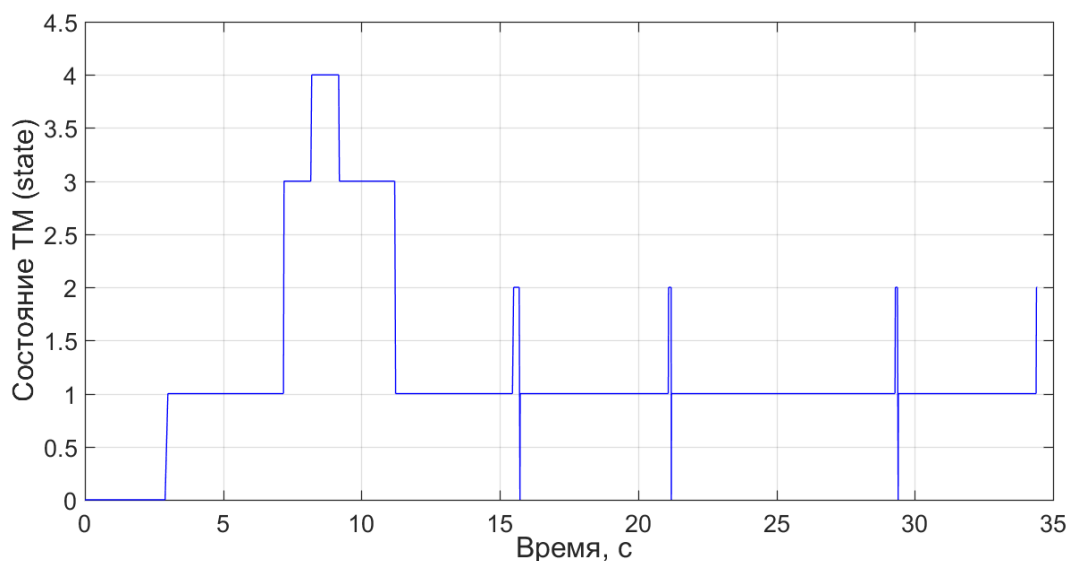


Рисунок 5.15. График изменения состояния ТМ в случае движения на карте с неподвижным препятствием

На графике видно, что примерно на 7 секунде произошло столкновение, и автомат ТМ изменил состояние «Moving_to_goal» (state = 1) на «Emergency» (state = 3). Спустя небольшое время произошёл переход в подсостояние «Back_away» (state = 4), в течение которого ТМ отъезжал назад. На графике также видно, что состояние «Idle» (state = 0) всегда занимает небольшой промежуток времени, поскольку после каждого прибытия в точку (state = 2), у автомата МИУ всегда есть новая цель, которую он отправляет с минимальной задержкой.

Рассмотрим похожую модель среды: на карте есть два коридора и МИУ строит два одинаковых пути к конечной цели (рисунок 5.16). При этом один из коридоров заблокирован *неподвижными* препятствиями, о которых не было известно в момент построения карты модулем интеллектуального управления. Пусть МИУ выбрал путь №1 с координатами целевых точек (0.75, 3), (4.25, 3) и (5, 2), и ТМ начинает последовательно посещать точки пути.

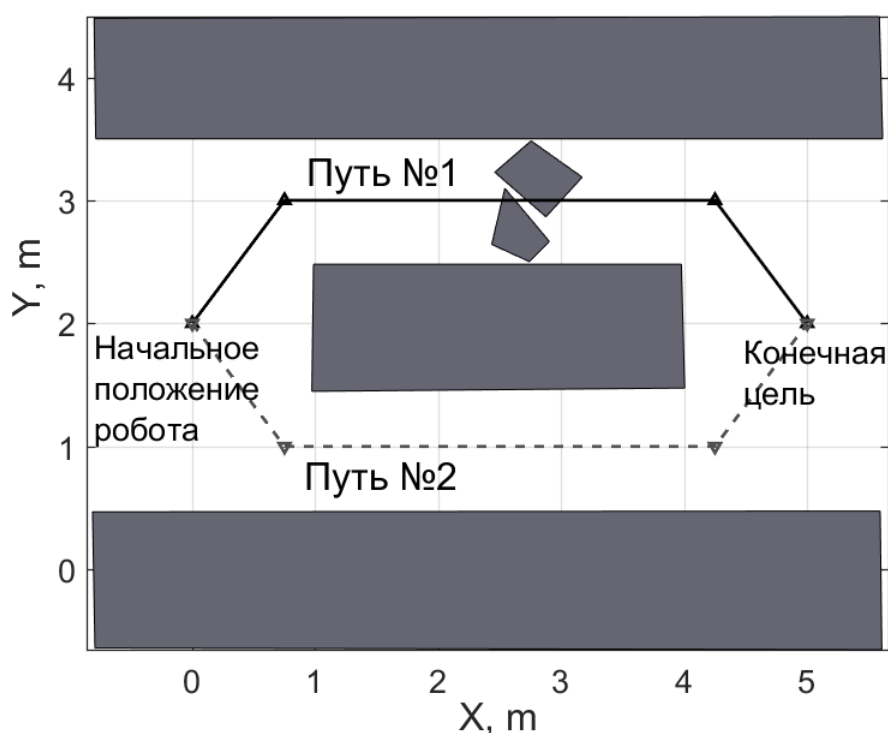


Рисунок 5.16. Модель среды с двумя коридорами, один из которых заблокирован

На рисунке 5.17 показана траектория робота после проведения компьютерной симуляции.

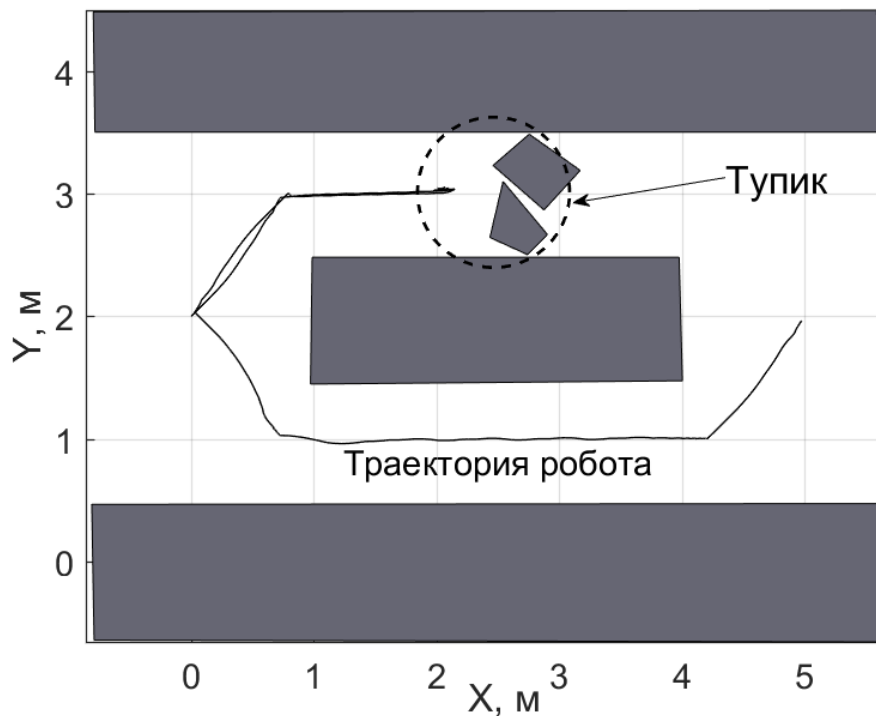


Рисунок 5.17. Траектория робота после проведения симуляции истечения времени прибытия к цели

ТМ посетил только одну точку первого пути. Далее препятствия в коридоре не позволили ТМ проехать к следующей цели, и он начал совершать небольшие движения в зоне тупика, пытаясь проехать вперёд. Напомним, что ТМ обладает информацией о среде только в *ближней зоне*, поэтому самостоятельно не может пройти через этот тупик. Перед началом симуляции было установлено предельное время прибытия к цели – $t_{\max} = 15$ с. Когда время на выполнение перемещения превысило $t_{\max}$, автомат ТМ перешёл в состояние «Emergency» и отправил отчёт в МИУ. Модуль интеллектуального управления зафиксировал экстренную ситуацию и переключился на второй путь с координатами целевых точек (0.75, 1), (4.25, 1), (5, 2), и ТМ снова начал перемещение. После этого робот успешно достиг конечной позиции с координатами (5, 2).

На графике изменения состояний автомата ТМ (рисунок 5.18) видно, что с момента достижения первой точки ($t = 7.4$ с) и до перехода в аварийный режим (22.4 с) прошло 15 с.

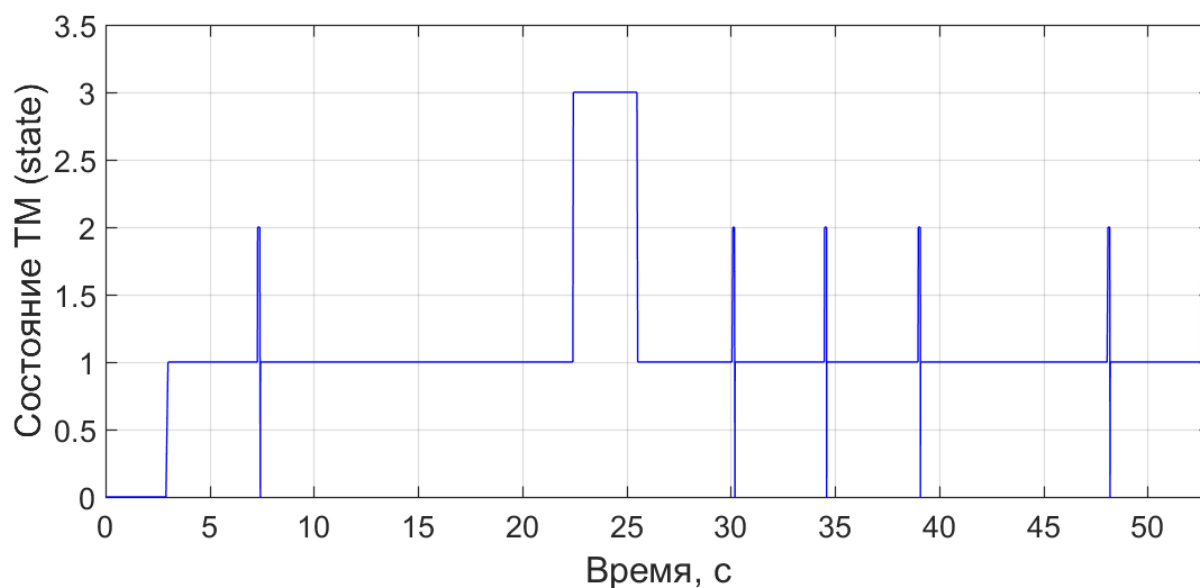


Рисунок 5.18. График изменения состояния автомата ТМ в случае движения на карте с коридорами

В аварийном режиме ТМ находился приблизительно 3.1 с. После смены пути ТМ посетил 5 точек, самое длительное перемещение заняло около 9 секунд – между точками (0.75, 1) и (4.25, 1).

5.3 Прототипы модулей и субмодулей мобильного робота

Для проведения тестов и отладки протокола информационного межмодульного взаимодействия были изготовлены: макет когнитивного субмодуля, прототип субмодуля-привода для управления коллекторным ДПТ, бесколлекторным ДПТ [2], силовой модуль, два прототипа колёсного транспортного модуля, а также так называемые базовые субмодули.

Макет *когнитивного субмодуля* ТМ состоит из отладочной платы STM32F407 Discovery и двух печатных плат (рисунок 5.19), на каждой из которых установлен драйвер TJA1050 для работы с шиной CAN и понижающий DC-DC преобразователь напряжения (для питания драйвера). Установленный на отладочной плате микроконтроллер STM32F407VGT6 имеет следующие основные характеристики: частота микропроцессора 168 МГц, 192 КБ ОЗУ, 1

Мбайт флеш-памяти, два интерфейса CAN и блок обработки чисел с плавающей точкой (FPU). Макет когнитивного субмодуля выполняет основные функции компьютерной модели ТМ (блок ТМ local planner, см. рисунок 5.5): расчёт скорости ТМ для объезда препятствий и движения к целевой точке (см. алгоритм 3.1), а также обмен информацией с субмодулями и модулями.

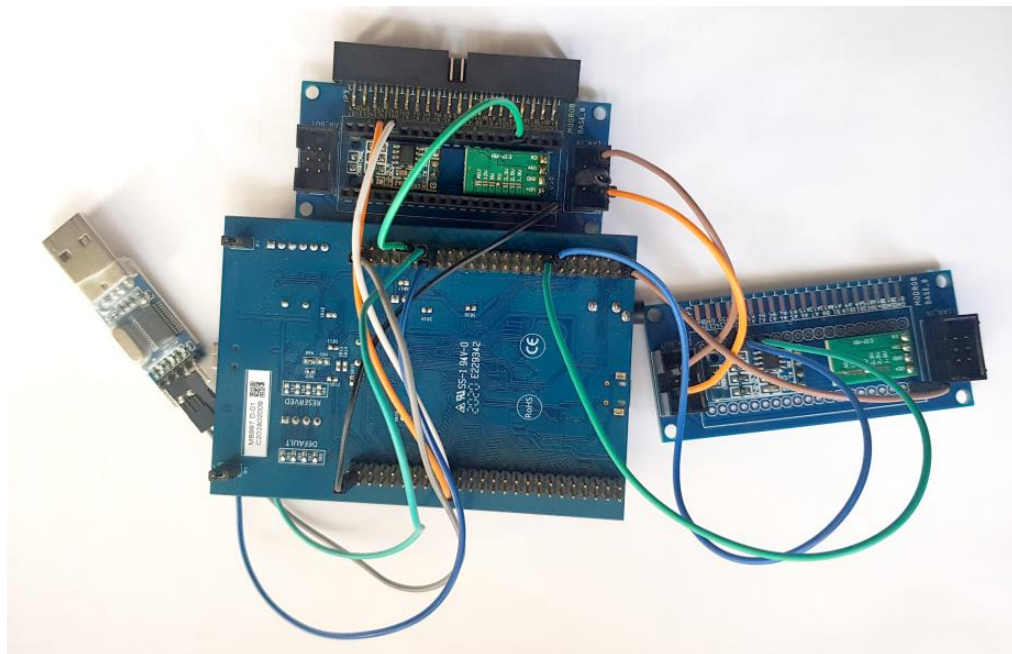


Рисунок 5.19. Макет когнитивного субмодуля

Код управляющей программы когнитивного субмодуля, а также других субмодулей выложен в открытой доступ на сайте Github [37]. Разработка управляющих программ модулей и субмодулей велась с использованием программного фреймворка для встраиваемых систем PlatformIO [44], а также инструментов и библиотек широко известного фреймворка STM32Cube [46].

Субмодуль-привод для управления коллекторным ДПТ: текущий прототип (рисунок 5.20) способен управлять двигателями с питающим напряжением от 12 до 24 В и номинальным током до 20 А. В качестве основного вычислителя используется микроконтроллер STM32F103C8T6 с ядром ARM-Cortex M3 с частотой 72 МГц, 20 КБ ОЗУ и 64 КБ флеш-памяти. Для работы с шиной CAN используется драйвер TJA1050. Для питания микроконтроллера и драйвера используется понижающий DC-DC преобразователь напряжения.

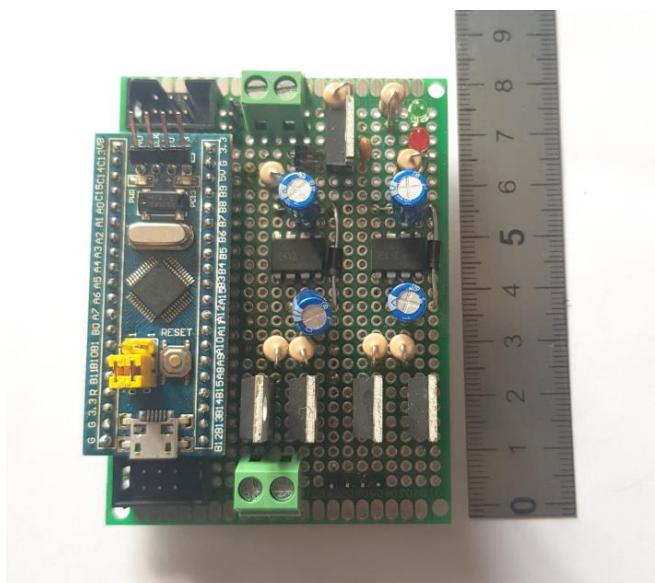


Рисунок 5.20. Прототип субмодуля управления коллекторным ДПТ

Замкнутое управление с обратной связью по положению вала и/или его скоростью реализуется, если к субмодулю ДПТ подключен энкодер. Субмодуль ДПТ публикует на шину информацию о скорости и положении вала с заданной частотой. Используя когнитивный субмодуль, можно настраивать те или иные параметры субмодуля ДПТ, например, коэффициенты ПИД-регулятора скорости.

Силовой модуль: текущий прототип модуля (рисунок 5.21) включает в себя литий-ионные аккумуляторы, два микроконтроллера ATtiny85, DC-DC преобразователь напряжения и вспомогательные электронные компоненты [1].



Рисунок 5.21. Прототип силового модуля (габаритные размеры: 140x100x90 мм)

Выходное напряжение модуля лежит в диапазоне 10-16.8 В, величина тока ограничена 5 А в продолжительном режиме, и 10 А – в кратковременном. В текущих экспериментах используются маломощные моторы, поэтому данные характеристики вполне удовлетворяют поставленным требованиям.

Для тестирования протокола межмодульной коммуникации и отладки алгоритмов управления субмодулей было разработано несколько так называемых базовых субмодулей.

Базовый субмодуль представляет собой электронное устройство, включающее в себя только основные компоненты: микроконтроллер STM32F103C8T6, драйвер CAN-шины и понижающий преобразователь напряжения DC-DC. Также на плате устройства имеются разъёмы для подключения к шине и вспомогательные электрические контакты, дублирующие выводы платы с микроконтроллером. На рисунке 5.22 приведена фотография четырёх изготовленных базовых субмодулей, подключенных к шине CAN.

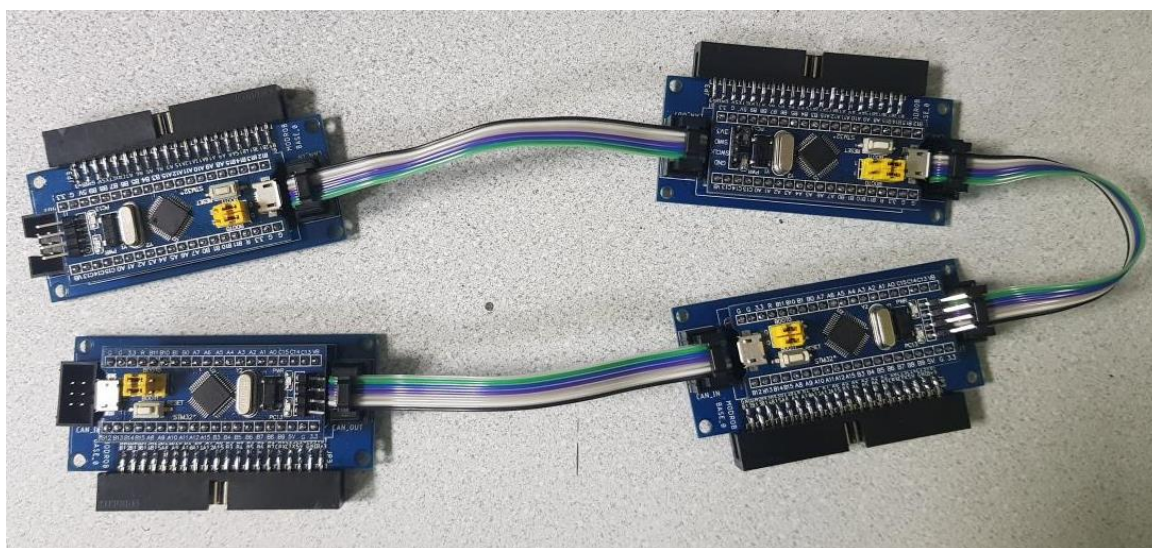


Рисунок 5.22. Базовые субмодули, подключённые к шине CAN с помощью шлейфа

Базовый субмодуль является *универсальным устройством*, способным выполнять функции остальных субмодулей при подключении необходимой периферии (например, силового преобразователя) и загрузки соответствующей управляющей программы.

Был изготовлен лабораторный макет транспортного модуля, представляющий собой колёсную платформу с дифференциальным приводом (рисунок 5.23).

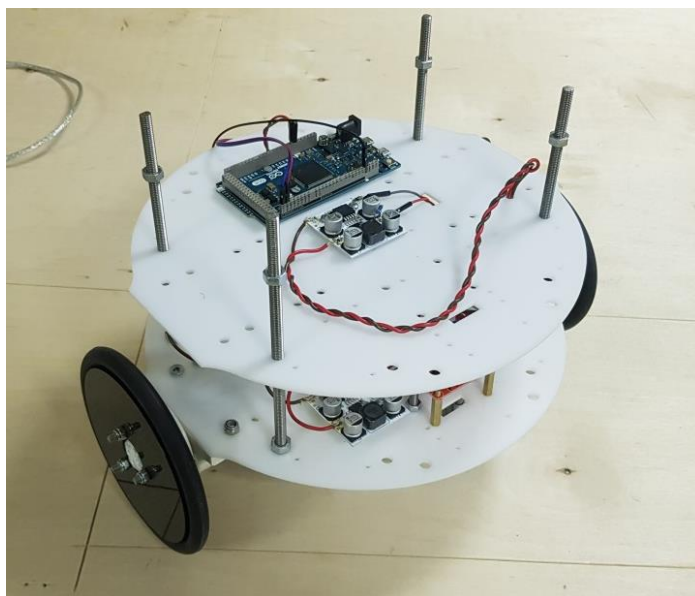


Рисунок 5.23. Лабораторный макет транспортного модуля с дифференциальным приводом

Аппаратная часть системы управления состоит из двух частей: задающей и управляющей (регулирующей). Конструктивно модуль также разделён на две части. В основе задающей части СУ ТМ (тактический уровень управления) лежит аппаратно-программная платформа Arduino Due с контроллером AT91SAM3X8E с ядром ARM и частотой процессора 84 МГц. За управляющую часть СУ (исполнительный уровень управления) отвечает аппаратно-программная платформа Arduino Mega 2560 с контроллером ATmega2560 и частотой 16 МГц.

Макет ТМ использовался для исследования алгоритма движения МР с использованием метода потенциальных полей [3], который рассматривался в разделе 3.3. Основной задачей платформы Arduino Due являлось вычисление заданной линейной и угловой скоростей ТМ в соответствии с предложенным алгоритмом движения. В свою очередь в платформе Arduino Mega было реализовано управление по скорости двумя ДПТ с помощью ПИД-регуляторов. Такое распределение вычислений соответствует модульной архитектуре ТМ, которая рассматривалась в разделе 3.6.

Был изготовлен лабораторный макет ТМ с всенаправленным мобильным механизмом, состоящим из платформы, четырёх ДПТ с энкодерами и механум-колёс (рисунок 5.24). В состав макета входят четыре субмодуля-привода для управления ДПТ и базовый субмодуль.

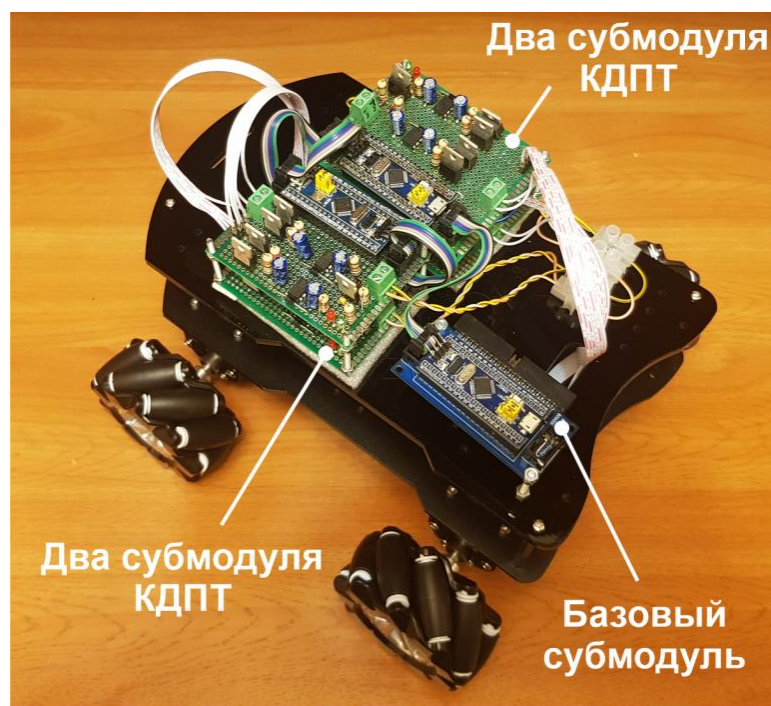


Рисунок 5.24. Лабораторный макет транспортного модуля с всенаправленным механизмом

5.4 Тестирование протокола ModRob

В этом разделе рассматривается эксперимент, в котором исследовалась возможность применения собственного протокола ModRob (см. раздел 4.1) для организации межмодульного информационного взаимодействия субмодулей ТМ в реальном времени.

Цель эксперимента: исследовать возможность использования разработанного протокола ModRob для реализации межмодульного информационного взаимодействия субмодулей в реальном времени. Схема эксперимента представлена на рисунке 5.25.

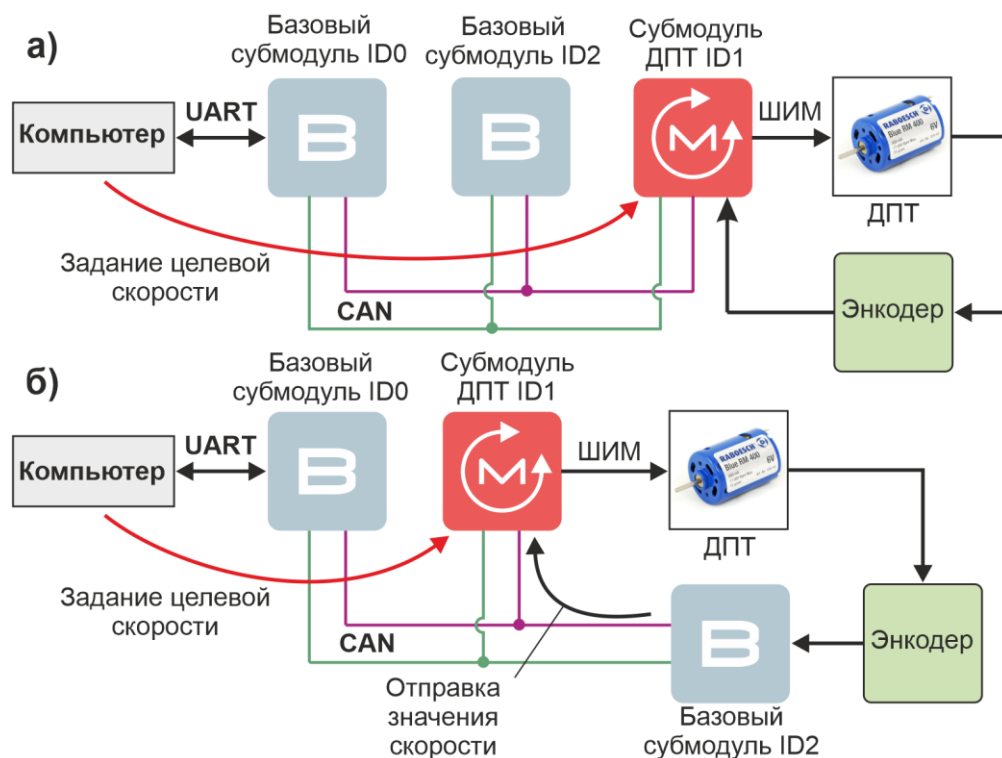


Рисунок 5.25. Схема экспериментальной установки: а) контур скорости субмодуля ДПТ замкнут на самом субмодуле ДПТ; б) контур скорости субмодуля ДПТ замкнут через промежуточный субмодуль с ID 2

Состав оборудования для проведения эксперимента включал в себя два базовых субмодуля и один субмодуль управления ДПТ (таблица 5.1).

Таблица 5.1. Состав оборудования для проведения эксперимента

Оборудование	Количество	Основные характеристики
Ноутбук (компьютер верхнего уровня)	1	Операционная система Manjaro Linux 19.0.0
Базовый субмодуль	3	См. раздел 5.3
Субмодуль ДПТ	1	См. раздел 5.3
Оптический инкрементальный энкодер	1	Минимальное расстояние между метками: 0.229 мм, напряжение питания: 5 В
Кодовый диск	1	Разрешающая способность: 30 имп/об
Электродвигатель постоянного тока Raboesch Blue RM-410	1	Номинальное напряжение питания: 7.2 В, номинальная скорость: 7500 об/мин, номинальный ток обмотки якоря 1.3 А, номинальный крутящий момент: 0.022 Нм

Постановка эксперимента. Была поставлена задача регулирования скорости вращения вала ДПТ с помощью субмодуля ДПТ. Измерение скорости вращения вала осуществлялось с помощью дифференцирования цифрового сигнала положения, который поступал с оптического энкодера.

Эксперимент был разделён на два опыта. В первом опыте (рисунок 5.25, а) электродвигатель с энкодером был подключен к субмодулю ДПТ (идентификатор ID 1), т.е. обратная связь по скорости замыкалась на этом же субмодуле. Во втором опыте энкодер подключался к другому субмодулю с ID 2 (рисунок 5.25, б), т.е. обратная связь по скорости замыкалась с помощью *промежуточного* субмодуля. Во всех опытах задание целевой скорости вращения вала ДПТ, а также установка частоты публикации переменных субмодулей и подписка переменных осуществлялась с помощью специальных команд, которые отправлялись с компьютера, подключенного к первому базовому субмодулю (ID 0) по интерфейсу UART. Подробное описание этих команд дано в статье [7].

Первый опыт. Субмодуль управления ДПТ измерял скорость вращения вала двигателя, дифференцируя сигнал с энкодера, и записывал её величину в переменную. С помощью соответствующей команды (таблица 4.1) была установлена частота публикации переменной на шину – 100 Гц. Привод субмодуля ДПТ должен был отрабатывать ступенчатые задающие воздействия, подаваемые на его вход: 100, 500, 800, 250, -250, -800, -500, -100 рад/с. Желаемая скорость вращения задавалась командой «Установка значения». В субмодуле ДПТ был реализован только контур скорости с классическим ПИД-регулятором⁶. В таблице 5.2 представлены исходные параметры лабораторной установки для первого и второго опытов.

Таблица 5.2. Исходные параметры лабораторной установки

Параметр	Значение
Пропускная способность CAN-шины, Мбит/с	1
Скорость работы последовательного порта (UART), бит/с	115200

⁶ ПИД-регулятор скорости не был настроен на оптимальный режим работы

Частота работы контура скорости в submodule ДПТ, Гц	100
Частота публикации переменной скорости на шину, Гц	100
Коэффициент усиления пропорциональной составляющей	3.5
Коэффициент усиления интегральной составляющей	9
Коэффициент усиления дифференциальной составляющей	0.2

Замечание: измеряемая скорость усредняется по 10 точкам.

Второй опыт. Как и в первом опыте, привод submodule ДПТ должен был обрабатывать ступенчатые задающие воздействия, подаваемые на его вход. Измерение скорости производилось в submodule с ID 2 (к которому подключен энкодер); публикация выполнялась с частотой 100 Гц. Для замыкания обратной связи submodule ДПТ был подписан на переменную скорости submodule с энкодером с помощью команды «Установка подписки».

На рисунке 5.26 штриховой линией показан график изменения задающего воздействия (желаемой скорости). Непрерывной линией показан график изменения скорости вращения вала электродвигателя, полученный в ходе первого опыта. Штрихпунктирной линией показан график изменения скорости вращения вала ДПТ, полученный в ходе второго опыта.

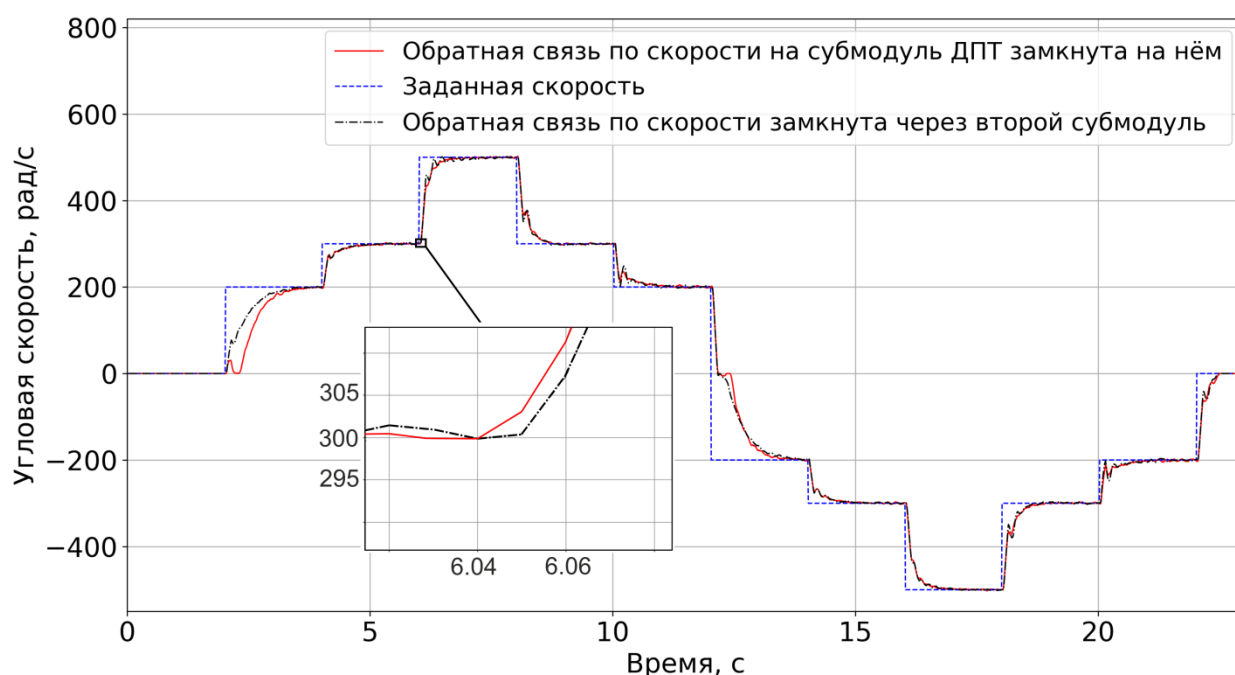


Рисунок 5.26. Сравнение скоростей вала ДПТ в первом и втором опыте

Из графиков видно, что качество переходных процессов изменения скорости во втором опыте практически не изменилось по сравнению с первым опытом, однако к задержке реакции привода на задающее воздействие добавилось запаздывание по сравнению с первым опытом. Запаздывание вызвано наличием промежуточного субмодуля в цепи обратной связи по скорости. Основные причины запаздывания: измерение скорости с периодом 10 мс, публикация с периодом 10 мс, а также обработка приёмником входящих сообщений, которая происходит с периодом 10 мс. Время, затрачиваемое на отправку сообщения по шине (порядка 14-16 мкс), вносит незначительный вклад в общую длительность запаздывания. В итоге запаздывание должно быть порядка 30 мс.

Анализ графиков показал, что реальное запаздывание варьируется в диапазоне от 4 мс до 50 мс, что соответствует расчёту (30 мс). В переходном процессе управления по скорости также появились незначительные колебания (в рамках установившейся ошибки в 5%) – по 2-3 полных колебания за время регулирования на скоростях более 200 рад/с.

Опыты были повторены для синусоидального задающего воздействия с частотой 0.5 Гц:

$$\omega_3 = 500 \cdot \sin(\pi t),$$

где ω_3 – заданная скорость, рад/с.

Все остальные параметры эксперимента остались прежними, кроме коэффициентов ПИД-регулятора скорости, значения которых были изменены: $K_P = 6$, $K_I = 20$ и $K_D = 0.3$.

На рисунке 5.27 показаны графики изменения скорости вращения вала двигателя для первого и второго опыта соответственно при синусоидальном задающем воздействии.

Как и в предыдущем случае, скорости в первом и во втором опытах отличаются незначительно, тем не менее, также присутствует задержка реакции привода. Задержка варьируется в диапазоне от 13 до 25 мс (в начальный момент времени). Средняя величина задержки – 20 мс.

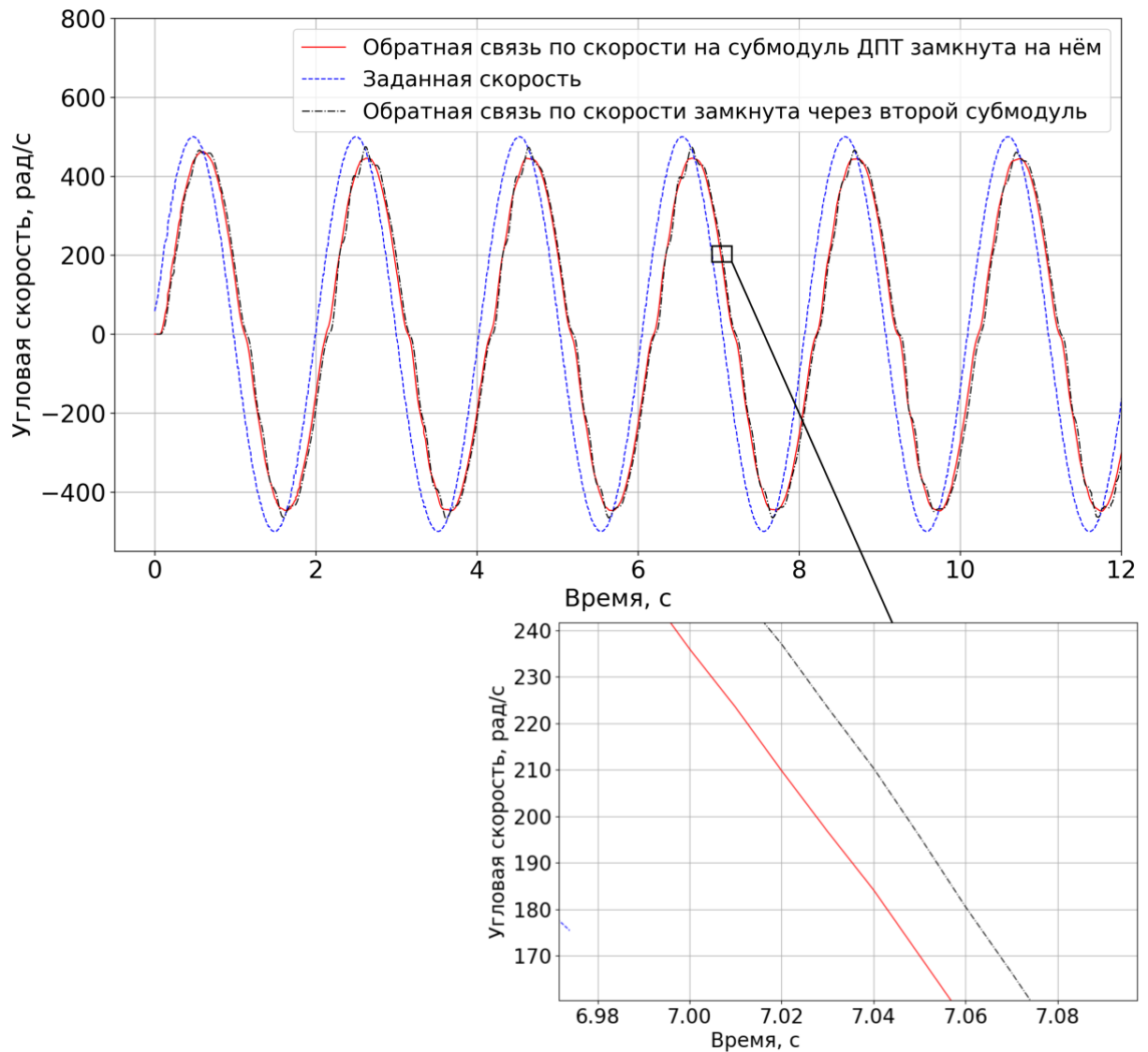


Рисунок 5.27. Сравнение скоростей ДПТ при подаче синусоидального воздействия

Выводы по эксперименту. Анализ результатов показал, что использование разработанного протокола ModRob *позволяет осуществлять управление субмодулями в реальном времени.*

Тем не менее, в силу определённых недостатков протокола и его малой распространённости было решено перейти к использованию протокола OpenCyphal, что было отмечено в разделе 4.1. Следующий эксперимент рассматривается для модулей и субмодулей, в которых реализован протокол OpenCyphal.

5.5 Экспериментальное исследование коммуникационных каналов модулей и субмодулей

Цель эксперимента: исследовать возможность работы ТМ в режиме реального времени при использовании разработанных типов данных для межмодульного взаимодействия с использованием протокола OpenCyphal (см. раздел 4.1) и алгоритма навигации в динамической среде (см. алгоритм 3.1). Для этого необходимо определить интервал времени от момента передачи сенсорных данных от СМБД к ТМ (когнитивному субмодулю) и до момента приёма сообщения от когнитивного субмодуля одним из базовых субмодулей при различных условиях.

Схема-эксперимента представлена на рисунке 5.28.

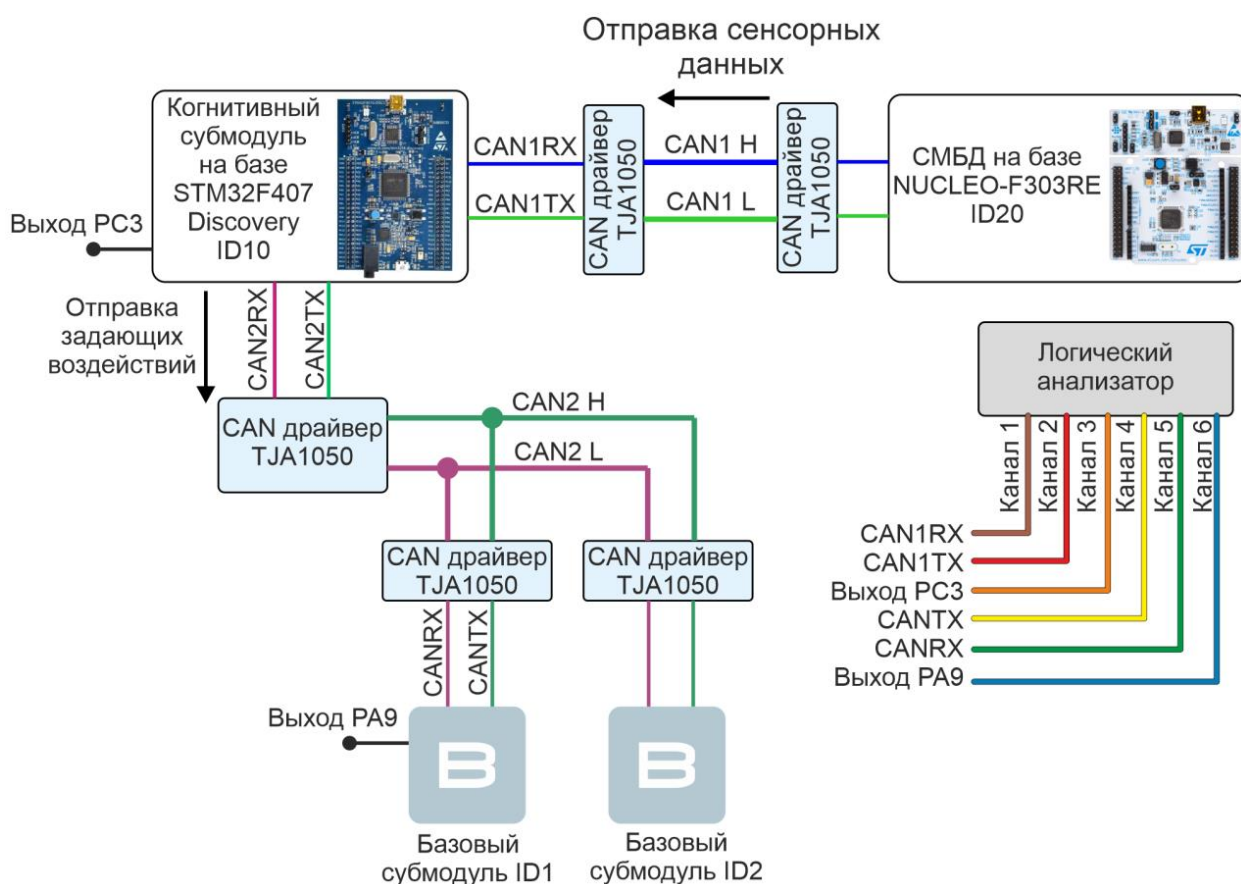


Рисунок 5.28. Схема эксперимента с протоколом OpenCyphal

Состав оборудования для проведения эксперимента:

1. Когнитивный субмодуль (см. раздел 5.3).

2. Макет сенсорного модуля ближнего действия, состоящий из отладочной платы NUCLEO-F303RE (микроконтроллер STM32F303RE), и печатной платы с драйвером шины CAN и понижающим преобразователем напряжения.

3. Два базовых субмодуля (см. раздел 5.3).

4. Цифровой анализатор Saleae Logic 16. Количество цифровых каналов – 16. Максимальная частота взятия отсчётов 100 МГц при использовании 3-х каналов, 50 МГц при использовании шести каналов. У анализатора два источника ошибок измерения длительности импульса:

- неточность частоты взятия отсчётов: ошибка 0.02% от длительности импульса (вносит вклад только для измерений импульсов с длительностью несколько секунд);
- асинхронное взятие отсчётов: ошибка $\pm \Delta t / f_{sample}$, где Δt – длительность импульса, f_{sample} – частота взятия отсчётов.

Версия ПО для работы с анализатором – Saleae Logic 1.2.18.

Описание эксперимента. В управляющей программе СМБД (NUCLEO-F303RE) было создано сообщение `modrob.sensor_module.sensor_data.0.2.uavcan` (см. раздел 4.1). Сообщение всегда включало в себя данные о 10 подвижных препятствиях и параметры отрезков прямых. При этом координаты центров препятствий были подобраны таким образом, чтобы они попадали в ближнюю зону робота (до 3 м). Тогда для каждого препятствия нужно вычислять область VO согласно алгоритму 3.1 (худший случай). Параметры отрезков прямых, описывающих статические препятствия, формировались с помощью генератора псевдослучайных чисел. Число отрезков прямых в эксперименте не превышало 90. Сенсорный модуль отправлял сообщение с сенсорными данными на шину CAN1 с частотой $f_{sensor} = 20$ Гц.

Когнитивный субмодуль ТМ был подписан на сообщения сенсорного модуля. Субмодуль принимал сообщения, извлекал из них полезные данные согласно спецификации протокола OpenCyphal и выполнял расчёт алгоритма объезда препятствий, используя данные о подвижных препятствиях и отрезках

прямых, описывающих статические препятствия. Длительность расчёта определялась с помощью измерения длительности импульса на выходе РСЗ платы Discovery: перед началом расчёта на выходе РСЗ устанавливалась логическая единица, после окончания расчёта – логический ноль.

После выполнения расчёта когнитивный субмодуль отправлял на шину CAN2 сообщение `modrob.transport_module.cognition_setpoint.0.1.uavcan`, которое содержало в себе заданные скорости и положения для субмодулей-приводов (в данном случае базовых субмодулей). Базовые субмодули были подключены к шине CAN2. На субмодуле с ID1 после успешного приёма сообщения `cognition_setpoint.0.1.uavcan` генерировался короткий импульс (выход PA9 базового субмодуля), который фиксировался с помощью цифрового анализатора.

Интервал времени ΔT с момента отправки сенсорных данных от СМБД к когнитивному субмодулю и до момента приёма сообщения когнитивного субмодуля одним из базовых субмодулей *должен быть меньше* периода отправки сенсорных данных $T_s = 50$ мс ($T_s = 1/f_{sensor}$). Соблюдение этого условия позволяет утверждать, что транспортный модуль функционирует в реальном времени.

Интервал времени ΔT включает в себя следующие составляющие (рисунок 5.29):

- задержка передачи сообщения от сенсорного модуля к когнитивному субмодулю – Δt_c ,
- длительность расчёта алгоритма навигации в когнитивном субмодуле транспортного модуля – Δt_a ,
- задержка передачи сообщения от когнитивного субмодуля к базовым субмодулям – Δt_0 .

Опыты показали, что задержка передачи сообщения от сенсорного модуля к когнитивному субмодулю (Δt_c) хорошо согласуется с теоретическим расчётом. Так для случая отправки сообщения СМБД, содержащего информацию о 10 подвижных препятствиях и 50 отрезках прямых, средняя величина задержки составляла 20.76 мс (при 20 замерах), а среднее квадратическое отклонение –

0.05 мс. Это соответствует теоретическому расчёту: размер сообщения СМБД с таким объёмом составляет 145 кадров CAN, длительность одного кадра – 144 мкс, тогда время передачи сообщения по шине составляет 20.88 мс.

На рисунке 5.29 показаны графики сигналов с линий CAN1RX, CAN1TX, выхода РСЗ (канал 3 анализатора), линий CANRX, CANTX (для анализа сообщений на шине CAN2) и выхода PA9 (канал 6 анализатора), полученные для одного из опытов с помощью логического анализатора.

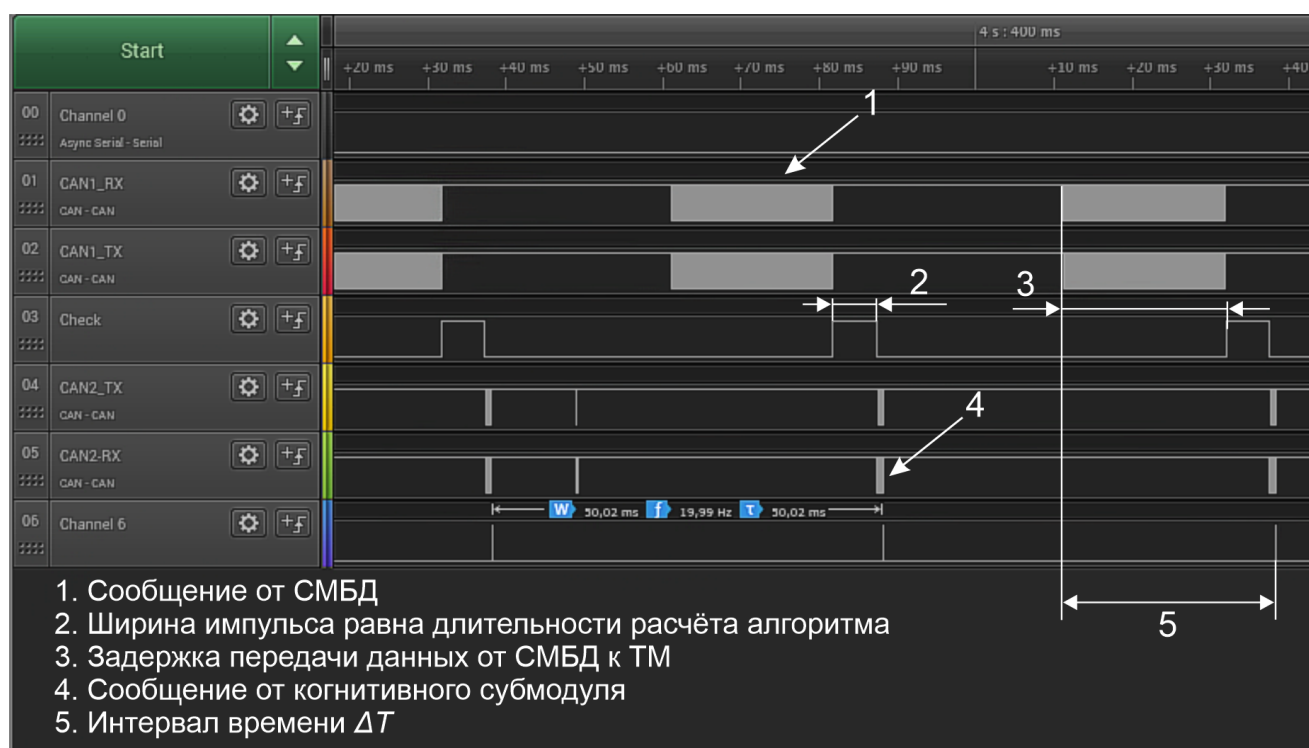


Рисунок 5.29. Графики сигналов, полученные для одного из опытов

Было выполнено несколько опытов по определению влияния количества отрезков прямых в сообщении от СМБД на длительность расчёта алгоритма навигации в когнитивном субмодуле (Δt_a) при фиксированном размере множества $RV - 100$ (число достижимых скоростей). Результаты опытов показаны в таблице 5.3. Все расчёты проводились по замерам длительностей 100 импульсов, которые генерировались на цифровом выходе РСЗ.

Таблица 5.3. Зависимость длительности расчёта алгоритма в когнитивном субмодуле от количества отрезков прямых

	Количество отрезков прямых				
	20	30	50	70	90
Минимальная длительность, мс	2.641	3.016	3.793	4.606	5.459
Максимальная длительность, мс	2.665	3.041	3.829	4.640	5.502
Средняя длительность, мс	2.652	3.027	3.810	4.621	5.475

Было выполнено несколько опытов по определению влияния размера множества достижимых скоростей RV на длительность выполнения расчёта алгоритма навигации. При этом количество отрезков прямых в сообщениях СМБД не менялось, и было равно 50. Результаты опытов приведены в таблице 5.4.

Таблица 5.4. Зависимость длительности расчёта алгоритма в когнитивном субмодуле от числа достижимых скоростей RV

	Число достижимых скоростей (размер RV)			
	64	100	144	169
Минимальная длительность, мс	2.525	3.793	5.349	6.246
Максимальная длительность, мс	2.546	3.829	5.415	6.331
Средняя длительность, мс	2.535	3.810	5.390	6.286

Видно, что длительность Δt_a расчёта алгоритма в транспортном модуле значительно зависит от объёма данных, которые отправляет СМБД, а также от реализации самого алгоритма (в частности, от размера RV). При этом важно найти компромисс между точностью и быстродействием. При большом размере области RV движение ТМ будет более точным и плавным, однако увеличенная задержка может привести к ухудшению реакции робота на внешние воздействия.

Была определена задержка передачи сообщения от когнитивного субмодуля к базовым субмодулям (Δt_0). Величина задержки зависит в основном от размера сообщения `cognition_setpoint.0.1.uavcan`, который в эксперименте не менялся и был равен 34 байтам. Анализ 20 замеров показал, что средняя задержка передачи

составляет 0.93 мс, среднее квадратическое отклонение – 0.01 мс. На рисунке 5.30 показано, как определяется величина задержки на графиках анализатора.

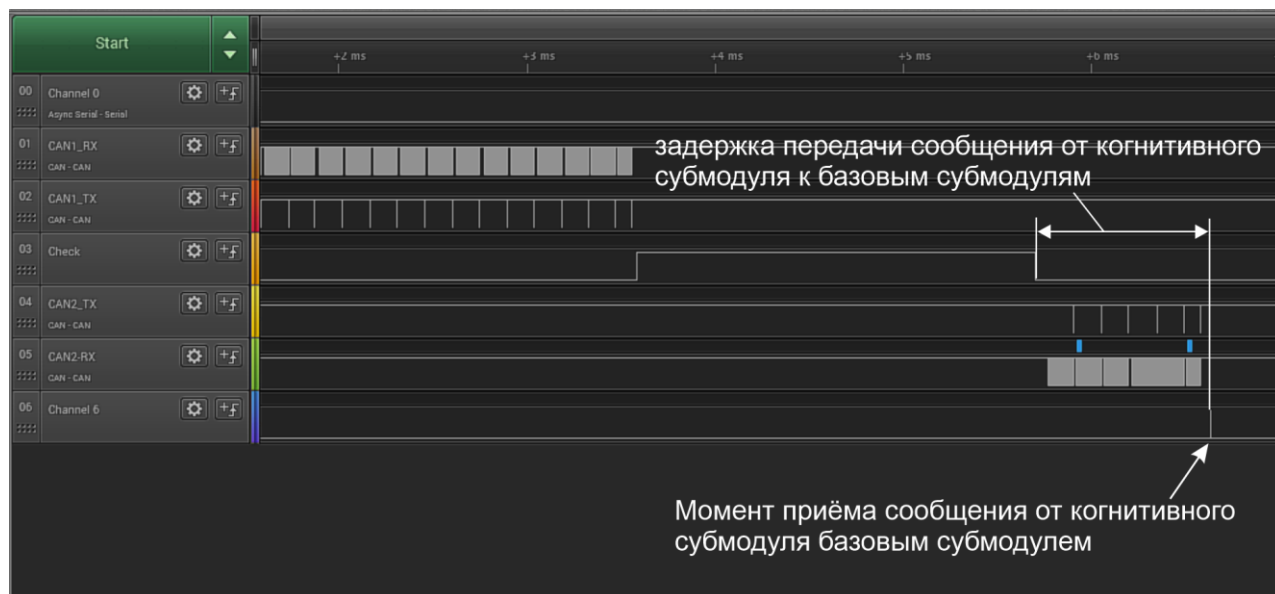


Рисунок 5.30. Задержка передачи сообщения от когнитивного субмодуля к базовым субмодулям

При известных составляющих интервала времени ΔT оценим его величину. Пусть СМБД отправляет сообщение, содержащее 90 отрезков прямых. Размер сообщения в кадрах CAN будет составлять 236 кадров, тогда $\Delta t_c = 33.98$ мс. При размере области RV равном 100, длительность расчёта алгоритма $\Delta t_a = 5.502$ мс (среднее значение из таблицы 5.3). При таких условиях интервал ΔT равен:

$$\Delta T = \Delta t_c + \Delta t_a + \Delta t_0 = 33.98 + 5.502 + 0.93 = 40.41 \text{ мс} < 50 \text{ мс}.$$

Значительный вклад в интервал ΔT вносит задержка передачи данных от СМБД к ТМ. Однако следует отметить, что эксперимент проводился с *классической* шиной CAN, в то время как предполагается, что полнофункциональные модули будут подключены к шине CAN FD, где скорость передачи кадров в несколько раз больше.

Рассчитаем интервал ΔT при условиях, которые соответствуют компьютерной модели (см. раздел 5.2). В этом случае в одном сообщении СМБД могут содержаться данные о 20-30 отрезках прямых, описывающих статические препятствия. Тогда для случая 30 отрезков интервал $\Delta t_c = 14.256$ мс. При размере

области RV равном 100 средняя длительность расчёта алгоритма $\Delta t_a = 3.027$ мс (см. таблицу 5.3), и интервал ΔT равен:

$$\Delta T = \Delta t_c + \Delta t_a + \Delta t_0 = 14.256 + 3.027 + 0.93 = 18.21 \text{ мс} < 50 \text{ мс}.$$

При этом в компьютерной модели задержка передачи данных от СМБД к приводам ТМ равна 50 мс. Таким образом, задержка, полученная экспериментально, оказалась более чем в два раза меньше задержки, заданной в компьютерной модели робота.

Вывод по эксперименту. Анализ результатов эксперимента показал, что система управления транспортного модуля работает в реальном времени: ТМ успевает принять сообщение от СМБД и выполнить все вычисления за интервал времени 50 мс и меньше. При этом быстроедействие ТМ и, соответственно, работа в реальном времени зависит от объёма сенсорных данных, которые принимает модуль, а также от реализации алгоритма навигации ТМ. Следует отметить, что в программном коде когнитивного субмодуля *не использовались* функции операционной системы реального времени (ОСРВ). При этом ОСРВ позволяют улучшить быстроедействие микроконтроллера за счёт многозадачности. Например, один вычислительный поток ОСРВ может отвечать за приём сообщений от СМБД, другой – за выполнение алгоритма навигации. В эксперименте все функции когнитивного субмодуля выполнялись в его программе последовательно в главном цикле, что называется режимом опроса (*polling*).

5.6 Выводы по пятой главе

Анализ результатов компьютерного моделирования показал работоспособность предложенного в главе 3 формального описания полнофункциональных модулей, а также алгоритма навигации в среде с подвижными и неподвижными препятствиями. Разработанная компьютерная модель отражает специфику работы мобильного робота с модульной архитектурой. С одной стороны, модули работают независимо: ТМ

самостоятельно перемещается к цели, объезжая препятствия, а МИУ выбирает путь на карте. С другой стороны, состояние одного модуля влияет на работу другого, что было показано в результатах компьютерной симуляции.

Тестирование протокола ModRob показало его применимость для межмодульного взаимодействия субмодулей ТМ в реальном времени. Задержка реакции привода, вызванная включением промежуточного субмодуля в цепь обратной связи контура управления приводом, не превышает 50 мс. При этом протокол OpenCyphal обладает рядом преимуществ по сравнению с ModRob и позволяет функционировать узлам сети в реальном времени, что было показано во втором эксперименте.

Эксперимент с когнитивным модулем ТМ и макетом сенсорного модуля позволил рассчитать интервал времени ΔT , в течение которого когнитивный субмодуль принимает сообщение от сенсорного модуля по шине CAN и вычисляет заданную скорость ТМ в соответствии с алгоритмом навигации. Показано, что интервал ΔT не превышает установленного порога в 50 мс, при следующих условиях (при учёте того, что для передачи сообщений от СМБД к ТМ использовалась обычная шина CAN, а не CAN FD):

- в одном сообщении от СМБД содержатся данные о не более чем 10 подвижных препятствиях и данные о не более чем 90 отрезках прямых, описывающих неподвижные препятствия;
- размер области достижимых скоростей RV не превышает 169.

При условиях, которые были заданы в компьютерной модели, интервал времени, полученный экспериментально, более чем в два раза меньше порога, установленного в модели, что подтверждает эффективность предложенных типов данных для сообщений и выполнения принципа полной функциональности.

ЗАКЛЮЧЕНИЕ

В ходе работы над диссертацией были получены следующие результаты:

1. Предложенный функционально-модульный подход к распределению функционала наземного МР на подфункции посредством иерархической многоуровневой архитектуры системы управления МР с использованием принципа полной функциональности обеспечивает такое распределение общего функционала МР между модулями и субмодулями на каждом уровне иерархии, при котором удаётся реализовать процесс управления на основе распределённых вычислений и тем самым решить проблему реального времени при усложнении функционала МР. Кроме того, за счёт упрощения алгоритмов СУ модулей и субмодулей их реализация может быть выполнена на основе встраиваемых систем.
2. Применение в технических разработках модулей и субмодулей принципа полной функциональности обеспечивает существенное снижение трафика межмодульного информационного взаимодействия на всех уровнях иерархии разработанной архитектуры. Данный результат позволяет для организации межмодульной коммуникации использовать простые и помехоустойчивые сети промышленного стандарта CAN, причём для уровня полнофункциональных модулей может использоваться модификация CAN FD.
3. Для реализации оперативного (режим «plug and play») реконфигурирования системы управления наземного МР должны соблюдаться требования сформулированного принципа полной функциональности, что позволяет использовать модули сторонних производителей посредством разработки соответствующих драйверов для каждого модуля и субмодуля.
4. Алгоритм информационного взаимодействия, разработанный на примере взаимодействия колёсного транспортного модуля с модулем

общесистемного (интеллектуального) управления, обеспечивает движение робота в среде с подвижными и неподвижными препятствиями. Для описания алгоритма использованы модели теории конечных автоматов для ТМ и МИУ при учёте требований к межмодульным интерфейсам ТМ и МИУ. Каждый автомат разработан с помощью диаграмм состояний UML, что позволило формально описать все основные функции модулей. Показано, что межмодульное взаимодействие в исследуемой архитектуре должно строиться на базе протокола OpenCyphal, который широко применяется в системах управления реального времени и удовлетворяет всем требованиям, сформулированным в процессе разработки собственного протокола ModRob.

5. Компьютерное моделирование систем управления основных функциональных модулей: МИУ, ТМ и СМБД доказало реализуемость предложенной архитектуры с заданными требованиями и условиями. Модули в компьютерной модели работают как самостоятельные агенты: транспортный модуль выполняет задание без постоянного контроля МИУ и требует его вмешательства только в случае экстренных ситуаций. Взаимодействие выполняется по схеме «ведущий-ведомый», что соответствует логике работы системы из конечных автоматов.
6. Экспериментальные исследования с использованием протоколов ModRob и OpenCyphal подтвердили возможность осуществления функционирования колёсного транспортного модуля в субмодульном исполнении в режиме реального времени. Задержки, вызванные затратами времени на передачу сообщений по шине CAN между модулями/субмодулями и выполнение вычислений на микроконтроллерах STM32F407VGT6, STM32F103C8T6 и STM32F303RE, находятся в допустимых пределах и соответствуют компьютерной модели и теоретическим расчётам.

Результаты диссертации были получены в ходе исследований, которые проводились при поддержке РФФИ: гранты №16-07-00811 А и №19-07-00892 А.

В дальнейшем следует выполнить исследования по взаимодействию модулей мобильного робота в реальных условиях с целью выявления новых свойств предложенной функционально-модульной архитектуры. Для этого планируется выполнить следующие работы:

- исследовать возможность оптимизации загруженности коммуникационной шины на уровне полнофункциональных модулей за счёт изменения полей структуры сенсорных данных, публикуемых сенсорным модулем ближнего действия;
- изготовить лабораторный макет модульного мобильного робота, состоящего из колёсного транспортного модуля, модуля интеллектуального управления и сенсорного модуля ближнего действия, и провести экспериментальные исследования с целью проверки предложенных решений в реальных условиях;
- провести эксперименты, подтверждающие эффективность использования предложенных методов по реконфигурированию колёсного мобильного робота с разработанной модульной архитектурой при выполнении задач в различных средах, отличающихся рельефом и окружающей обстановкой.

СПИСОК ЛИТЕРАТУРЫ

1. Абдразаков Е.Г. Энергообеспечение модульного мобильного робота в комплексе с коммуникационным каналом по линиям электропитания / Е.Г. Абдразаков, В.П. Андреев // Робототехника и техническая кибернетика. – 2021. – Т. 9. – № 1. – С. 59-71.
2. Андреев В.П. Разработка субмодуля управления бесколлекторным двигателем постоянного тока / В.П. Андреев, В.Л. Ким, Ю.Е. Луцкий // Экстремальная робототехника (ЭР-2021). Труды 32-й международной научно-технической конференции. – СПб.: ЦНИИ РТК, 2021. – С. 87-88.
3. Андреев В.П. Метод управления движением модульного мобильного робота с использованием двумерных векторных полей / В.П. Андреев, В.Л. Ким // Робототехника и техническая кибернетика. – 2017. – № 4(17). – С. 22-27.
4. Андреев В.П. Принцип полной функциональности модулей в гетерогенных модульных мобильных роботах / В.П. Андреев, В.Л. Ким, П.Ф. Плетенев // Экстремальная робототехника (ЭР-2017). Труды международной научно-технической конференции. – СПб.: ЦНИИ РТК, 2017. – С. 81-91.
5. Андреев В.П. Сетевые решения в архитектуре гетерогенных модульных мобильных роботов / В.П. Андреев, В.Л. Ким, Ю.В. Подураев // Робототехника и техническая кибернетика. – 2016. – № 3(12). – С. 23-29.
6. Андреев В.П. Разработка функциональных узлов гетерогенного модульного мобильного робота / В.П. Андреев, В.Л. Ким // Экстремальная робототехника (ЭР-2016). Труды международной научно-технической конференции. – СПб.: ООО «АП4Принт», 2016. – С. 359-369.
7. Андреев В.П. Аппаратно-программный фреймворк для разработки модульных мобильных роботов с иерархической архитектурой / В.П. Андреев, В.Л. Ким, С.Р. Эприков // Известия ЮФУ. Технические науки. Раздел IV. Связь, навигация и наведение. – 2020. – № 1. – С. 199-218.
8. Андреев В.П. Метод информационного взаимодействия для систем распределённого управления в роботах с модульной архитектурой / В.П. Андреев, П.Ф. Плетенев // Труды СПИИРАН. – 2018. – № 2 (57). – С. 134-160.
9. Андреев В.П. Разработка спецификации коммуникационного протокола для модульного мобильного робота ModRob-1 / В.П. Андреев, П.Ф. Плетенев, С.Р. Эприков // Сборник материалов XV Всероссийской научно-практической конференции «Перспективные системы и задачи управления». – Нижний Архыз: Издательство Южного федерального университета, 2020. – С. 391-405.
10. Андреев В.П. Функционально-модульный принцип построения гетерогенных мобильных роботов / В.П. Андреев, Ю.В. Подураев //

Экстремальная робототехника (ЭР-2016). Труды международной научно-технической конференции. – СПб.: ООО «АП4Принт», 2016. – С. 44-49.

11. Брауэр В. Введение в теорию конечных автоматов. Радио и связь / В. Брауэр; ред. Ю.И. Журавлёв; пер. К.В. Рудаков. – М., 1987. – 392 с.

12. Васильев А.В. Уточнение типоразмерных групп наземных дистанционно управляемых машин для применения в опасных для человека условиях / А.В. Васильев, А.В. Лопота // Научно-технические ведомости СПбГПУ. – 2015. – № 1(214). – С. 226-234.

13. Герасимов В.Н. Система навигации сервисного робота в среде с динамическими препятствиями : дис. ... кандидата техн. наук: 05.02.05, защищена 2016 / В.Н. Герасимов. – ФГБОУ ВО «МГТУ им. Н.Э. Баумана», 2016. – 187 с.

14. ГОСТ 15971-90 Системы обработки информации. Термины и определения. – М.: Издательство стандартов, 1991. – 12 с.

15. ГОСТ Р 60.0.0.2-2016 Роботы и робототехнические устройства. Классификация. – М.: Стандартинформ, 2018. – 10 с.

16. ГОСТ Р 60.0.0.4-2019 Роботы и робототехнические устройства. Термины и определения. – М.: Стандартинформ, 2019. – 26 с.

17. ГОСТ Р 60.0.0.5-2019 Роботы и робототехнические устройства. Мобильные роботы. Термины и определения. – М.: Стандартинформ, 2019. – 14 с.

18. ГОСТ Р 60.6.3.1-2019 Роботы и робототехнические устройства. Методы испытаний сервисных мобильных роботов для работы в экстремальных условиях. Термины и определения. – М.: Стандартинформ, 2019.

19. Ермолов И.Л. Расширение функциональных возможностей мобильных технологических роботов путём повышения уровня их автономности с использованием иерархической комплексной обработки бортовых данных : автореф. дис. ... д-ра техн. наук: 05.02.05: защищена 2012 / И.Л. Ермолов. – М.: ФГБОУ ВО «МГТУ «СТАНКИН», 2012. – 41 с.

20. «КАПИТАН» - малогабаритный робототехнический комплекс (МРК) [Электронный ресурс]. – URL: <https://rtc.ru/solution/kapitan/> (дата обращения: 05.02.2022).

21. Корчак В.Ю. Перспективы развития наземных робототехнических комплексов военного и специального назначения / В.Ю. Корчак, В.С. Лапшов, И.В. Рубцов // Известия ЮФУ. Технические науки. – 2015. – № 10(171). – С. 83-95.

22. Кульгина С.П. Система дистанционного управления для робототехнических комплексов трёх сред базирования / С.П. Кульгина, А.Н. Косенко, Д.С. Попов // Экстремальная робототехника (ЭР-2021). Труды 32-й международной научно-технической конференции. – СПб.: ЦНИИ РТК, 2021. – С. 205-215.

23. «КУРСАНТ» - малогабаритная разведывательная платформа (МРП) [Электронный ресурс]. – URL: <https://rtc.ru/solution/kursant/> (дата обращения: 17.05.2022).
24. Лопота А.В. Этапы и перспективы развития модульного принципа построения робототехнических систем / А.В. Лопота, Е.И. Юревич // Научно-технические ведомости СПбГПУ. Информатика. Телекоммуникации. Управление. – 2013. – № 1. – С. 98-103.
25. Образовательные комплекты и полигоны ЦНИИ РТК - образовательный набор «КОНСТРУКТОР» [Электронный ресурс]. – URL: <https://rtc.ru/solution/konstruktor/> (дата обращения: 26.04.2022).
26. Павловский В.Е. Модульная микроконтроллерная система управления роботами РОБОКОН-1 / В.Е. Павловский, В.В. Павловский. – Препринты ИПМ им. М. В. Келдыша. – ИПМ им. М. В. Келдыша, 2012. – 32 с.
27. Платонов А.К. Робототехника лунной базы / А.К. Платонов // XXXIV Чтения по космонавтике. – ИПМ им. М.В. Келдыша РАН, 2010.
28. Подураев Ю.В. Мехатроника: основы, методы, применение: учеб. пособие для студентов вузов / Ю.В. Подураев. – 2-е изд., стер. – М.: Машиностроение, 2007. – 256 с.
29. Попов Е.П. Основы робототехники: введение в специальность / Е.П. Попов, Г.В. Письменный. – М.: Высшая школа, 1990. – 224 с.
30. Словарь по кибернетике / ред. В.С. Михалевич. – 2-е. – Киев: Главная редакция Украинской Советской Энциклопедии имени М. П. Бажана, 1989. – 751 с.
31. Таненбаум Э. Архитектура компьютера : Классика Computer science / Э. Таненбаум, Т. Остин. – 6-е. – СПб.: Питер, 2013. – 816 с.
32. Таненбаум Э. Распределенные системы. Принципы и парадигмы : Классика Computer Science / Э. Таненбаум, М. ван Стеен. – Санкт-Петербург: Питер, 2003. – 880 с.
33. Управляемый досмотровый робототехнический комплекс «Скарабей» версия 2.0 «М» [Электронный ресурс]. – URL: https://www.set-1.ru/products/detail.php?ELEMENT_ID=140 (дата обращения: 26.04.2022).
34. Юревич Е.И. Основы робототехники : Учебная литература для вузов / Е.И. Юревич. – 3-е изд., перераб. и доп. – СПб.: БХВ-Петербург, 2010. – 368 с.
35. Юревич Е.И. Роботы ЦНИИ РТК на Чернобыльской АЭС и развитие экстремальной робототехники : Монография / Е.И. Юревич. – СПб.: Изд-во Политехнического ун-та, 2004.
36. CANaerospace - the Airborne CAN Interface Standard [Электронный ресурс]. – URL: <https://www.stockflightsystems.com/canaerospace.html> (дата обращения: 12.03.2022).

37. Kim V. Modrob-UAVCAN project [Электронный ресурс]. – URL: <https://github.com/ValeriiKim/Modrob-UAVCAN> (дата обращения: 12.03.2022).
38. Kirienko P. Cyphal/CAN: CAN FD bandwidth estimation [Электронный ресурс]. – URL: https://docs.google.com/spreadsheets/d/1iK0MegMuEC55czTW6ssWhrA_sGUuSIT0S26xv5gytE/edit#gid=0 (дата обращения: 16.05.2022).
39. Kirienko P. Cyphal/CAN: Classic CAN bandwidth estimation [Электронный ресурс]. – URL: <https://docs.google.com/spreadsheets/d/1zjpdPfmBf9oje2qjLYddlhFfkaQuTJ-VvQjHxzITils/edit#gid=0> (дата обращения: 16.05.2022).
40. Kirienko P. Nunavut: DSDL transpiler [Электронный ресурс]. – URL: <https://github.com/OpenCyphal/nunavut> (дата обращения: 30.03.2022).
41. Kirienko P. OpenCyphal: an open technology for robust real-time intravehicular distributed computing and communication based on modern networking standards. OpenCyphal: robust intravehicular distributed computing / P. Kirienko, S. Dixon, others Published: Internet. – 2014.
42. Leading open-source components for UAVs [Электронный ресурс]. – URL: <https://www.dronecode.org/projects/> (дата обращения: 12.03.2022).
43. Levi Z. Automotive Ethernet: The Future of In-Car Networking? [Электронный ресурс]. – URL: <https://www.electronicdesign.com/markets/automotive/article/21806349/automotive-ethernet-the-future-of-incar-networking> (дата обращения: 12.03.2022).
44. Professional collaborative platform for embedded development [Электронный ресурс]. – URL: <https://platformio.org/> (дата обращения: 14.03.2022).
45. SAE J1939 Standards Collection on the Web: Content [Электронный ресурс]. – URL: <https://www.sae.org/standardsdev/groundvehicle/j1939a.htm> (дата обращения: 12.03.2022).
46. STM32Cube Ecosystem [Электронный ресурс]. – URL: https://www.st.com/content/st_com/en/ecosystems/stm32cube-ecosystem.html (дата обращения: 14.03.2022).
47. Unified Modeling Language (UML) Version 2.5.1 / S. Cook [и др.]. – Object Management Group (OMG), 2017.
48. A comparison of line extraction algorithms using 2D range data for indoor mobile robotics / V. Nguyen [et al.] // Autonomous Robots. – 2007. – Vol. 23. – № 2. – P. 97-111.
49. A modular architecture for a multi-purpose mobile robot / G. Steinbauer [et al.] // Innovations in Applied Artificial Intelligence. IEA/AIE 2004. Lecture Notes in Computer Science. – 2004. – Vol. 3029.

50. A Modularized Personal Robot DRP I: Design and Implementation / S. Roh [et al.] // IEEE Transactions on Robotics. – 2009. – Vol. 25. – A Modularized Personal Robot DRP I. – № 2. – P. 414-425.
51. Acutronic Robotics fails to find funding for H-ROS for robot hardware [Электронный ресурс]. – URL: <https://www.therobotreport.com/acutronic-robotics-h-ros-robot-hardware-fails/> (дата обращения: 27.05.2021).
52. Ahmadzadeh H. Modular Robotic Systems: Characteristics and Applications / H. Ahmadzadeh, E. Masehian, M. Asadpour // Journal of Intelligent & Robotic Systems. – 2016. – Vol. 81. – Modular Robotic Systems. – № 3-4. – P. 317-357.
53. AMiRo: a modular & customizable open-source mini robot platform / S. Herbrechtsmeier [et al.] // 20th International Conference on System Theory, Control and Computing (ICSTCC). – Sinaia: IEEE, 2016. – P. 687-692.
54. Andreev V. Control system and design of the motion module of a heterogeneous modular mobile robot / V. Andreev, V. Kim // Proceedings of the 27th International DAAAM Symposium on Intelligent Manufacturing and Automation. – Mostar, Bosnia and Herzegovina: DAAAM International, 2016. – P. 0586-0594.
55. Andreev V. ModRob: the Hardware-Software Framework for Modular Mobile Robots Prototyping / V. Andreev, V. Kim, S. Eprikov // Proceedings of the 31st DAAAM International Virtual Symposium. – Mostar, BiH: DAAAM International, 2020. – P. 0391-0402.
56. Andreev V. Formal specification of a mobile robot with modular architecture for navigation in dynamic environment / V. Andreev, V. Kim // Proceedings of the 32nd International DAAAM Virtual Symposium on Intelligent Manufacturing and Automation. – Vienna, Austria: DAAAM International, 2021. – P. 0215-0227.
57. Andreev V. The principle of full functionality – the basis for rapid reconfiguration in heterogeneous modular mobile robots / V. Andreev, V. Kim, P. Pletenev // Proceedings of the 28th International DAAAM Symposium on Intelligent Manufacturing and Automation. – Zadar, Croatia: DAAAM International, 2017. – P. 0023-0028.
58. Andreev V. A hierarchical modular architecture for reconfigurable mobile robots / V. Andreev, V. Kim, V. Pryanichnikov // Proceedings of the 30th International DAAAM Symposium on Intelligent Manufacturing and Automation. – Zadar, Croatia: DAAAM International, 2019. – P. 1152-1159.
59. Andreev V. Problems of Choosing an Intermodule Information Interaction Protocol for Mobile Robots with Modular Control System Architecture / V. Andreev, P. Pletenev // Proceedings of the 32nd International DAAAM Virtual Symposium on Intelligent Manufacturing and Automation. – Vienna, Austria: DAAAM International, 2021. – P. 0151-0157.
60. Andreev V. Network-based Design of Heterogeneous Modular Mobile Robotic Systems / V. Andreev, Y. Poduraev // Proceedings of the 27th International DAAAM

Symposium on Intelligent Manufacturing and Automation. – Mostar, Bosnia and Herzegovina: DAAAM International, 2016. – P. 0004-0009.

61. ASEBA: A Modular Architecture for Event-Based Control of Complex Robots / S. Magnenat [et al.] // IEEE/ASME Transactions on Mechatronics. – 2011. – Vol. 16. – ASEBA. – № 2. – P. 321-329.

62. Autonomous mobile robot with multiprocessor control system / S. Szabo [et al.] // Proceedings of 7th International Workshop on Advanced Motion Control. – Maribor, Slovenia: IEEE, 2002. – P. 467-471.

63. Azevedo J.L. Hierarchical Distributed Architectures for Autonomous Mobile Robots: a Case Study / J.L. Azevedo, B. Cunha, L. Almeida // IEEE Conference on Emerging Technologies and Factory Automation (EFTA 2007). – Patras, Greece: IEEE, 2007. – P. 973-980.

64. Berg J. van den. Reciprocal Velocity Obstacles for real-time multi-agent navigation / J. van den Berg, Ming Lin, D. Manocha // 2008 IEEE International Conference on Robotics and Automation 2008 IEEE International Conference on Robotics and Automation (ICRA). – Pasadena, CA, USA: IEEE, 2008. – P. 1928-1935.

65. Borenstein J. The vector field histogram-fast obstacle avoidance for mobile robots / J. Borenstein, Y. Koren // IEEE Transactions on Robotics and Automation. – 1991. – Vol. 7. – № 3. – P. 278-288.

66. CANopen – The standardized embedded network [Электронный ресурс]. – URL: <https://www.can-cia.org/canopen/> (дата обращения: 27.05.2021).

67. Chen Y. RRT* Combined with GVO for Real-time Nonholonomic Robot Navigation in Dynamic Environment / Y. Chen, M. Liu // arXiv:1710.07102 [cs]. – 2018.

68. Cherubini A. Autonomous Visual Navigation and Laser-Based Moving Obstacle Avoidance / A. Cherubini, F. Spindler, F. Chaumette // IEEE Transactions on Intelligent Transportation Systems. – 2014. – Vol. 15. – № 5. – P. 2101-2110.

69. ClearPath: highly parallel collision avoidance for multi-agent simulation / S.J. Guy [et al.] // Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation - SCA '09. – New Orleans, Louisiana: ACM Press, 2009. – ClearPath. – P. 177-187.

70. Collision avoidance under bounded localization uncertainty / D. Claes [et al.] // 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems. – Vilamoura-Algarve, Portugal: IEEE, 2012. – P. 1192-1198.

71. Decentralized non-communicating multiagent collision avoidance with deep reinforcement learning / Y.F. Chen [et al.] // 2017 IEEE International Conference on Robotics and Automation (ICRA). – Singapore: IEEE, 2017. – P. 285-292.

72. Design of reconfigurable heterogeneous modular architecture for service robots / Ho Seok Ahn [et al.] // 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems. – Nice: IEEE, 2008. – P. 1313-1318.

73. Design of the ATRON lattice-based self-reconfigurable robot / E.H. Østergaard [et al.] // *Autonomous Robots*. – 2006. – Vol. 21. – № 2. – P. 165-183.
74. Design of transmute: A modular self-reconfigurable robot with versatile transformation capabilities / G. Qiao [et al.] // *2012 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. – Guangzhou, China: IEEE, 2012. – Design of transmute. – P. 1331-1336.
75. Distributed and Modular CAN-Based Architecture for Hardware Control and Sensor Data Integration / D.P. Losada [et al.] // *Sensors*. – 2017. – Vol. 17. – № 5. – P. 1013-1030.
76. Fiorini P. Motion Planning in Dynamic Environments Using Velocity Obstacles / P. Fiorini, Z. Shiller // *The International Journal of Robotics Research*. – 1998. – Vol. 17. – № 7. – P. 760-772.
77. Fox D. The dynamic window approach to collision avoidance / D. Fox, W. Burgard, S. Thrun // *IEEE Robotics & Automation Magazine*. – 1997. – Vol. 4. – № 1. – P. 23-33.
78. Fujita M. Digital creatures for future entertainment robotics / M. Fujita // *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation*. – San Francisco, CA, USA: IEEE, 2000. – P. 801-806.
79. Fukuda T. A dynamically reconfigurable robotic system (concept of a system and optimal configurations) / T. Fukuda, S. Nakagawa // *IECON '87: Industrial Applications of Robotics & Machine Vision*. – Cambridge, Massachusetts: SPIE-the Society of Photo-optical Instrumentation Engineers, 1987.
80. Harel D. Statecharts: a visual formalism for complex systems / D. Harel // *Science of Computer Programming*. – 1987. – Vol. 8. – № 3. – P. 231-274.
81. Hart P.E. A formal basis for the heuristic determination of minimum cost paths / P.E. Hart, N.J. Nilsson, B. Raphael // *IEEE Transactions on Systems Science and Cybernetics*. – 1968. – Vol. 4. – № 2. – P. 100-107.
82. HEBI Robotics [Электронный ресурс]. – URL: <https://www.hebirobotics.com/> (дата обращения: 27.05.2021).
83. Hendzel Z. Modelling of dynamics of a wheeled mobile robot with mecanum wheels with the use of lagrange equations of the second kind / Z. Hendzel, L. Rykala // *Int. J. of Applied Mechanics and Engineering*. – 2017. – Vol. 22. – № 1. – P. 81-99.
84. IEEE Standard for Robot Map Data Representation for Navigation. – IEEE, 2015. – 54 p.
85. Independent navigation of multiple mobile robots with hybrid reciprocal velocity obstacles / J. Snape [et al.] // *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. – St. Louis, MO, USA: IEEE, 2009. – P. 5917-5922.
86. ISO/DIS 22166-1:2021(E). – 2021.

87. Jahn U. Concepts of a Modular System Architecture for Distributed Robotic Systems / U. Jahn, C. Wolff, P. Schulz // Computers. – 2019. – Vol. 8. – № 1. – P. 25.
88. Karaman S. Sampling-based algorithms for optimal motion planning / S. Karaman, E. Frazzoli // The International Journal of Robotics Research. – 2011. – Vol. 30. – № 7. – P. 846-894.
89. LaValle S.M. Rapidly-exploring random trees: A new tool for path planning / S.M. LaValle. – Computer Science Dept.: Iowa State University, 1998.
90. Lightweight Collision Avoidance for Resource-Constrained Robots / M. Shahriari [et al.] // 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). – Madrid: IEEE, 2018. – P. 1-9.
91. Minguez J. Nearness diagram (ND) navigation: collision avoidance in troublesome scenarios / J. Minguez, L. Montano // IEEE Transactions on Robotics and Automation. – 2004. – Vol. 20. – № 1. – P. 45-59.
92. Modeling Software with Finite State Machines: A Practical Approach / F. Wagner [et al.]. – New York: Taylor & Francis Group, LLC, 2006. – 370 p.
93. Modular design: a plug and play approach to sensory modules, actuation platforms, and task descriptions for robotics and automation applications / A. Elkady [et al.] // Journal of Intelligent & Robotic Systems. – 2014. – № 75. – P. 271-289.
94. Modular Mobile Robotic Kit for Prototyping and Debugging of Control Algorithms / O. Shmakov [et al.] // Proceedings of the 28th International DAAAM Symposium on Intelligent Manufacturing and Automation. – Zadar, Croatia: DAAAM International, 2017. – P. 0950-0956.
95. M-TRAN: self-reconfigurable modular robotic system / S. Murata [et al.] // IEEE/ASME Transactions on Mechatronics. – 2002. – Vol. 7. – M-TRAN. – № 4. – P. 431-441.
96. Myon, a New Humanoid / M. Hild [et al.] // Language Grounding in Robots / eds. L. Steels, M. Hild. – Boston, MA: Springer US, 2012. – P. 25-44.
97. Narayanan V. Anytime Safe Interval Path Planning for dynamic environments / V. Narayanan, M. Phillips, M. Likhachev // 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems. – Vilamoura-Algarve, Portugal: IEEE, 2012. – P. 4708-4715.
98. Optimal Reciprocal Collision Avoidance for Multiple Non-Holonomic Robots : Springer Tracts in Advanced Robotics / J. Alonso-Mora [et al.] // Distributed Autonomous Robotic Systems. – 2013. – Vol. 83. – P. 203-216.
99. Pavlidis T. Segmentation of Plane Curves / T. Pavlidis, S.L. Horowitz // IEEE Transactions on Computers. – 1974. – Vol. C-23. – № 8. – P. 860-870.
100. Phidgets Inc. - Products for USB Sensing and Control [Электронный ресурс]. – URL: <https://www.phidgets.com/> (дата обращения: 27.05.2021).

101. R2P: An open source hardware and software modular approach to robot prototyping / A. Bonarini [et al.] // *Robotics and Autonomous Systems*. – 2014. – Vol. 62. – № 7. – P. 1073-1084.
102. *Robotics Modelling, Planning and Control : Advanced Textbooks in Control and Signal Processing* / B. Siciliano [et al.]. – Springer-Verlag London, 2009. – 632 p.
103. ROS - Robot Operating System [Электронный ресурс]. – URL: <https://www.ros.org/> (дата обращения: 27.05.2021).
104. Sahin F. Groundscouts: architecture for a modular micro robotic platform for swarm intelligence and cooperative robotics / F. Sahin // *IEEE International Conference on Systems, Man and Cybernetics*. – The Hague, Netherlands: IEEE, 2004. – P. 929-934.
105. Saranli U. A modular real-time fieldbus architecture for mobile robotic platforms / U. Saranli, A. Avci, M.C. Öztürk // *IEEE Transactions on Instrumentation and Measurement*. – 2011. – Vol. 60. – № 3. – P. 916-927.
106. Savkin A.V. A simple biologically inspired algorithm for collision-free navigation of a unicycle-like robot in dynamic environments with moving obstacles / A.V. Savkin, C. Wang // *Robotica*. – 2013. – Vol. 31. – № 6. – P. 993-1001.
107. Siegwart R. *Introduction to Autonomous Mobile Robots : Intelligent Robotics and Autonomous Agents series* / R. Siegwart, I. Nourbakhsh, D. Scaramuzza. – second edition. – The MIT Press, 2011. – 472 p.
108. Stentz A. Optimal and Efficient Path Planning for Partially-Known Environments / A. Stentz // *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*. – San Diego, CA, USA, 1994. – Vol. 4. – P. 3310-3317.
109. Stentz A. The Focussed D* Algorithm for Real-Time Replanning / A. Stentz // *Proceedings of the International Joint Conference on Artificial Intelligence*. – Montreal Quebec, Canada, 1995. – Vol. 2. – P. 1652-1659.
110. The Hybrid Reciprocal Velocity Obstacle / J. Snape [et al.] // *IEEE Transactions on Robotics*. – 2011. – Vol. 27. – № 4. – P. 696-706.
111. The shift in the robotics paradigm — The Hardware Robot Operating System (H-ROS); an infrastructure to create interoperable robot components / V. Mayoral [et al.] // *2017 NASA/ESA Conference on Adaptive Hardware and Systems (AHS)*. – Pasadena, CA, USA: IEEE, 2017. – P. 229-236.
112. Wang D. MEMS Mirrors for LiDAR: A Review / D. Wang, C. Watkins, H. Xie // *Micromachines*. – 2020. – Vol. 11. – MEMS Mirrors for LiDAR. – № 5. – P. 456-480.