

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский политехнический университет» («Московский Политех»)

На правах рукописи

Алексеев Александр Александрович

СИСТЕМА ПЛАНИРОВАНИЯ И ИСПОЛНЕНИЯ ЗАДАНИЯ МОБИЛЬНЫМ
КОЛЁСНЫМ РОБОТОМ НА ОСНОВЕ МЕТОДА ОПОРНЫХ ТРАЕКТОРИЙ И
КРИВЫХ БЕЗЪЕ ТРЕТЬЕГО ПОРЯДКА В УСЛОВИЯХ СКЛАДА
МАШИНОСТРОИТЕЛЬНОГО ПРЕДПРИЯТИЯ

Специальность: 05.02.05 – Роботы, мехатроника и робототехнические системы
(технические науки)

диссертация на соискание ученой степени

кандидата технических наук

Научный руководитель:
кандидат технических наук, доцент
Палагута Константин Алексеевич

Москва – 2019

Введение.....	4
Глава 1. Автономный транспортный модуль в условиях логистики склада машиностроительного предприятия.....	10
1.1 Динамика внедрений робототехнических решений в складскую логистику	10
1.2 Логистические АТМ, примеры внедрения, лидеры отрасли.....	14
1.3 Перспективные решения в развитии логистических АТМ.....	19
1.4 Логистические АТМ. Рабочая зона, техническое оснащение, перемещаемые грузы	27
1.5 Краткая классификация методов планирования маршрута АТМ.....	31
1.6 Интеграция нескольких методов планирования маршрута АТМ.....	37
1.7 Выводы по главе 1.....	39
Глава 2. Кинематическое управление движением четырёхколёсных АТМ	40
2.1 Скоростные режимы движения логистических АТМ.....	40
2.2 Рулевое управление Аккермана. Кинематика поворота четырёхколёсного транспортного средства.....	41
2.3 Математическая модель кинематического управления	48
четырёхколёсным ТС.....	48
2.4 Выводы по главе 2.....	55
Глава 3. Разработка алгоритмического обеспечения системы формирования маршрута колёсного робота	56
3.1 Представления мобильной платформы в работе	58
3.2 Опорная траектория. Кривые Безье	61
3.3 Алгоритм исполнения задания	66
3.4 Алгоритмическое обеспечение модуля генератора опорной траектории.....	74
3.5 Формирование исходного пространства поиска.....	76
3.5 Формирование конечного пространства поиска.....	80
3.6 Выбор кривой опорной траектории из конечного множества кривых, удовлетворяющих задаче перемещения	82

3.7 Выводы по главе 3.....	85
Глава 4. Экспериментальное исследование разработанного алгоритмического обеспечения системы управления колёсным роботом	87
4.1 Исполнение задания, парковка робота в ограниченном пространстве	87
4.2 Исполнение задания, траектория ухода от столкновения с препятствием	90
4.3 Поиск опорной траектории, уход от столкновения с препятствием	96
4.4 Полунатурные испытания, отработка траектории.....	115
4.5 Экспериментальное исследование зависимости «плавности» хода мобильной платформы от количества пролётных точек в задании.....	119
4.6 Сравнительная оценка временных затрат, связанных с перемещением грузов	125
4.7 Выводы по главе 4.....	127
Заключение	129
Список сокращений и условных обозначений.....	131
Список терминов	132
Список литературы	134
Приложение А	149
Приложение Б	163
Приложение В.....	167

Введение

Актуальность темы. В настоящий момент времени разработан целый ряд роботов, направленных на автоматизацию процессов транспортировки, учёта и хранения грузов в условиях складских комплексов машиностроительных предприятий, а также крупных логистических центров. Однако, несмотря на это, существует широкий круг задач, где обработка грузов по сей день осуществляется в «ручном режиме». Во-первых, это участок «первоначальной приёмки», на котором происходит разгрузка объектов, прибывших на хранение, во-вторых, участок транспортировки груза от места «первоначальной приёмки» к пункту «учёта и сортировки», где перемещение грузов осуществляется посредством различного рода транспортёров под управлением человека-оператора. Следует заметить, что вопрос автоматизации участка «первоначальной приёмки» не имеет чисто технического решения, т.к. требует наличия согласованных стандартов хранения и размещения перемещаемых объектов при транспортировке между складским предприятием и поставщиком продукции. В то же время вопрос исключения человека-оператора из процесса перемещения грузов по территории логистических центров является актуальной технической задачей, т.к. автоматизация данного участка складского комплекса позволит увеличить скорость обработки грузов, что в свою очередь приведёт к увеличению «пропускной способности» логистического центра в целом. Помимо этого, автоматизация данного участка транспортировки, во-первых, позволит исключить ошибки на этапе конечного хранения, обусловленные «человеческим фактором» (груз доставлен не на ту полку/ленту), во-вторых, обеспечит максимально эффективное использование всех имеющихся в наличии транспортёров, исключая их простой или «балластные» перемещения.

Следует заметить, что сегодня существует множество технических средств, успешно решающих задачи автоматизации внутрицеховых перевозок. Однако в силу особенностей условий, в которых осуществляется транспортировка объектов хранения по территории логистических центров, робототехнические устройства,

получившие широкое распространение при проектировании производственных линий машиностроительных предприятий, в данном случае не применимы, что показано в работе [103], и обусловлено подходом к организации системы управления траекторий движения внутрицеховых автоматических самоходных тележек – АСТ. Таким образом, одной из основных задач, которая должна быть решена в рамках реализации робототехнической системы транспортировки объектов хранения на заявленном участке складского комплекса, является задача планирования маршрутов и исполнения заданий мобильными роботами на территории логистического центра, учитывающая особенности рабочего пространства, в котором функционирует робот-транспортёр.

Разработанность темы. На текущий момент над решением задач, направленных на создание системы автоматической транспортировки объектов хранения по территории складского комплекса, работает целый ряд ведущих промышленных предприятий и университетов западной Европы. В рабочую группу проекта Plug & Navigate robots for smart factories (PAN-Robots) входят: Aikio M., Beinschob, P., Boehning M., Cotena G., Digani V., Estefania J.A, Fantuzzi C., Fuerstenberg K., Golban C., Kregel A., Magnani M., Marita T., Mäkinen J.T., Nagy A.E., Nedevschi S., Ramos B., Reinke C., Reinke C., Ronzoni D., Sabbatini L., Secchi C., Stimming C., Szakats I., Yang B., опубликовавшие по данной тематике более 60 печатных работ (с полным списком публикаций можно ознакомиться по адресу <http://www.pan-robots.eu/deliverables/publications/> [дата доступа: 24.11.2017]). Однако значительная часть деталей реализации проекта не разглашается.

В частности, недостаточно освящён вопрос планирования маршрутов выполнения задания автономными транспортными модулями – АТМ, в то время как, в контексте задачи создания робототехнического комплекса для автоматизации транспортировки объектов хранения по территории склада машиностроительного предприятия вопрос разработки алгоритмов планирования маршрутов АТМ является одним из ключевых и требует детальной проработки.

Цель работы. Целью диссертационной работы является минимизация затрат времени на перемещение объектов складирования по территории логистического центра посредством разработки группы алгоритмов автоматического управления траекторий движения роботов-транспортёров.

Задачи исследования. Достижение поставленной цели предполагает последовательное решение следующих задач:

1. Провести анализ существующих подходов к реализации систем формирования и исполнения задания колёсными мобильными роботами.
2. Разработать алгоритм формирования маршрутов колёсного мобильного робота, учитывающий особенности конструкции мобильной платформы и рабочего пространства, в котором функционирует робот-транспортёр.
3. Разработать алгоритм выбора маршрута выполнения задания из множества альтернативных вариантов.
4. Разработать алгоритм и программное обеспечение, обеспечивающее расчёт управляющих воздействий, необходимых для исполнения задания мобильным роботом;
5. Произвести контрольные испытания и апробацию разработанных алгоритмов формирования маршрута и исполнения задания мобильным роботом посредством вычислительных и полунатурных экспериментов.

Объектом исследования является автономный транспортный модуль робототехнической системы транспортировки объектов складирования по территории складского комплекса.

Предметом исследования являются алгоритмы определения маршрута задания для мобильных платформ, трансмиссия которых имеет четырёхколёсную компоновку автомобильного типа.

Научная новизна заключается в следующем:

1. Установлена связь между траекторией движения мобильного робота и кривыми Безье третьего порядка, основанная на особенности конструкции мобильных платформ, имеющих четырёхколёсную компоновку автомобильного типа.

2. Разработан алгоритм планирования задания мобильного робота, особенностью которого является использование кривых Безье третьего порядка для определения маршрутов задания.

3. Разработан алгоритм выбора маршрута выполнения задания мобильным роботом из множества альтернативных маршрутов, заданных кривыми Безье третьего порядка, особенностью которого является применение эволюционных методов поиска.

4. Разработан алгоритм определения пролётных точек маршрута заданного кривой Безье третьего порядка для мобильных платформ, трансмиссия которых имеет четырёхколёсную компоновку автомобильного типа.

5. Разработан алгоритм автоматического расчёта набора управляющих воздействий, необходимых для отработки маршрута заданного кривой Безье третьего порядка, для мобильных платформ, трансмиссия которых имеет четырёхколёсную компоновку автомобильного типа.

Теоретическая значимость работы заключается в разработке алгоритмов, использующих кривые Безье третьего порядка, для формирования маршрута выполнения задания колёсным роботом, трансмиссия мобильной платформы которого имеет четырёхколёсную компоновку автомобильного типа.

Практическая значимость заключается в:

- Разработанной компьютерной программе, которая может быть использована в качестве инструмента настройки и отладки системы планирования маршрута исполнения задания при конструировании колёсных роботов, имеющих мобильную платформу, с трансмиссией реализованной на базе четырёхколёсной компоновки автомобильного типа.

- В возможности применения разработанного программного обеспечения и аппаратной базы в учебном процессе подготовки студентов высших учебных

заведений в рамках лабораторного практикума по программированию систем управления мобильными роботами.

Методы исследования. При решении поставленных задач в работе использована теория кривых Безье, эволюционные методы оптимизации, генетические алгоритмы. Экспериментальное исследование выполнялось посредством вычислительных и полунатурных экспериментов. Моделирование процессов транспортировки грузов по территории логистического центра осуществлялось с применением программного пакета интеллектуального имитационного моделирования RDO-STUDIO.

На защиту выносятся:

1. Алгоритм планирования задания мобильного робота, использующий кривые Безье третьего порядка для задания маршрута.
2. Алгоритм определения пролётных точек маршрута мобильного робота, заданного кривой Безье третьего порядка.
3. Алгоритм выбора маршрута выполнения задания мобильным роботом из множества альтернативных маршрутов, определённых кривыми Безье третьего порядка.
4. Алгоритм автоматического расчёта набора управляющих воздействий (углов отклонения управляющих колёс мобильного робота от продольной оси мобильной платформы), необходимых для отработки мобильным роботом маршрута, заданного кривой Безье третьего порядка.

Степень достоверности и апробация результатов работы. Достоверность и обоснованность научных результатов диссертационной работы обусловлена применением известной и получившей широкое распространение теории кривых Безье и подтверждается согласованием результатов теоретических и практических исследований. Достоверность и новизна полученных результатов научной работы защищена тремя патентами.

Основные положения диссертационной работы были доложены и обсуждены:

- на заседаниях аттестационной комиссии при ежегодной аттестации аспирантов кафедры «Автоматика, информатика и системы управления» ГОУ МГИУ;
- на конференции молодых ученых на кафедре «Автомобили и двигатели» МГИУ, г. Москва – 2009 г;
- на научно-практической конференции «Инновации на основе информационных и коммуникационных технологий», ИНФО г. Сочи – 2010 г;
- на IV Международной научно-практической конференции «Информационные технологии в образовании, науке и производстве», г. Серпухов – 2010 г;
- на Научно-практической конференции студентов, аспирантов и молодых учёных Автомобильного факультета МГИУ «Молодая наука АФ» – 2011 г.;
- на научно-практической конференции «ИНФО-2015: Инновации на основе информационных и телекоммуникационных технологий» – 2015 г;
- на научно-практической конференции «ИНФО-2016: Информационные и коммуникационные технологии в образовании, науке и производстве» – 2016 г.

Соответствие работы паспорту специальности. Диссертационная работа соответствует научной специальности 05.02.05 «Роботы, мехатроника и робототехнические системы» в пп. 2 и 3 паспорта специальности.

Публикации. Основное содержание диссертационной работы опубликовано в 20 печатных работах, из них 6 – в журналах из перечня рецензируемых научных изданий, в которых должны быть опубликованы основные научные результаты диссертаций на соискание учёной степени кандидата наук, на соискание учёной степени доктора наук, рекомендованного ВАК, и 3 патента.

Структура и объём диссертационной работы, Диссертационная работа состоит из введения, четырёх глав, заключения, списка литературы и трёх приложений, содержит 75 рисунков и 52 таблиц. Объем диссертации 185 страниц. Список литературы состоит из 134 наименования.

Глава 1. Автономный транспортный модуль в условиях логистики склада машиностроительного предприятия.

1.1 Динамика внедрений робототехнических решений в складскую логистику

Внедрение робототехнических комплексов на производстве началось с 1960-х годов прошлого столетия с появления первого серийного промышленного робота «Юнимейт», производства фирмы «Юнимейшн» [2]. Процесс роботизации приобрёл масштабный характер к середине 80-х годов прошлого столетия и продолжается по настоящее время. Так, например, за последние 4 года (2016 - 2019), количество отгрузок промышленных роботов в Мире показывает уверенный рост в среднем на 13,5% процентов ежегодно (рисунок 1.1) [3].

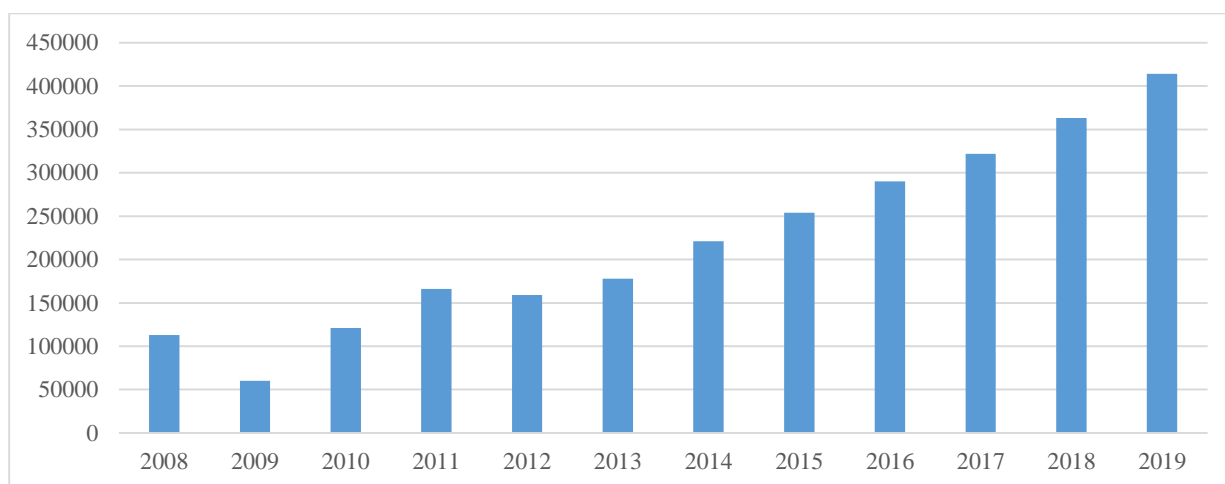


Рисунок 1.1 – Ежегодные отгрузки промышленных роботов в Мире в период с 2008 по 2019 год. Рисунок выполнен по материалам публикации [3]

Причём общий уровень промышленной автоматизации, который выражается в количестве роботов на каждые 10000 работников, также демонстрирует уверенный рост. Лидером промышленной автоматизации является Южная Корея, в которой на 2016 год на каждые 10000 рабочих насчитывается 631 робот, при среднем мировом показателе 74 робота на каждые 10000 работников [4]. Общий характер

показателя роботизации промышленности по странам приведён на рисунках 1.2 и 1.3.

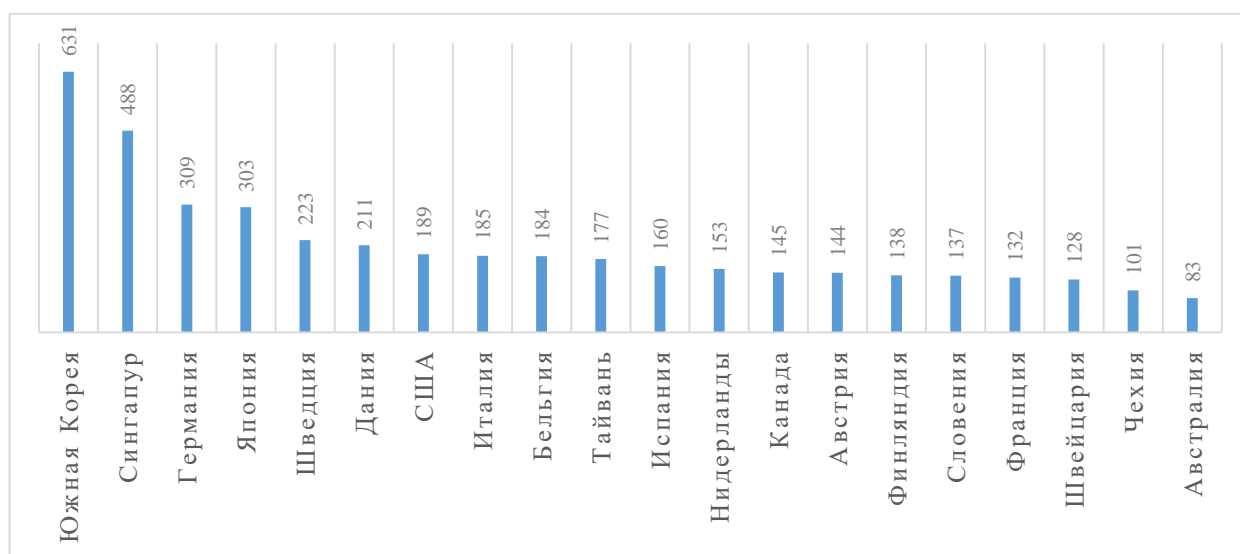


Рисунок 1.2 – Количество установленных промышленных роботов на 10000 рабочих в промышленности на 2016 год. По материалам публикаций [3, 4]

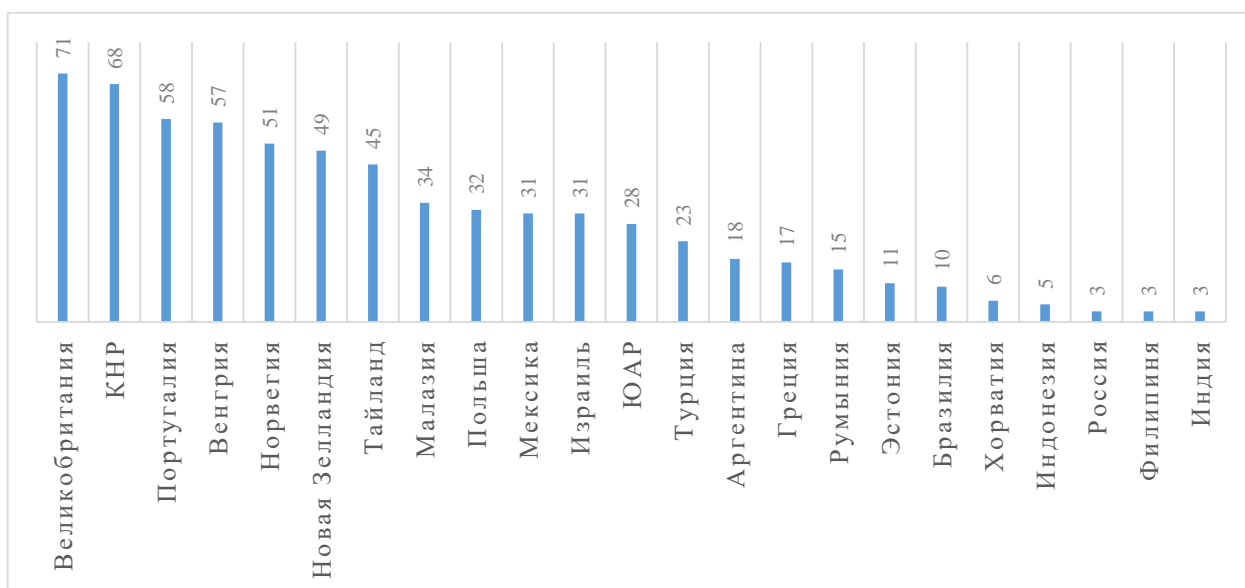


Рисунок 1.3 – Количество установленных промышленных роботов на 10000 рабочих в промышленности на 2016 год. По материалам публикаций [3, 4]

Несмотря на то, что склад [5] является важной и неотъемлемой частью любого производственного процесса, автоматизация складского хозяйства промышленных предприятий долгое время являлась «узким» местом в создании автоматизированных производств [6].

Так, например, даже для складских комплексов категории А+, по международной классификации Кнайт Франк (Knight Frank) [7] до 2008 года существовал проблемный с точки зрения автоматизации участок транспортировки объектов складирования между зоной приёма/отгрузки и зоной хранения, рисунок 1.4 [6].

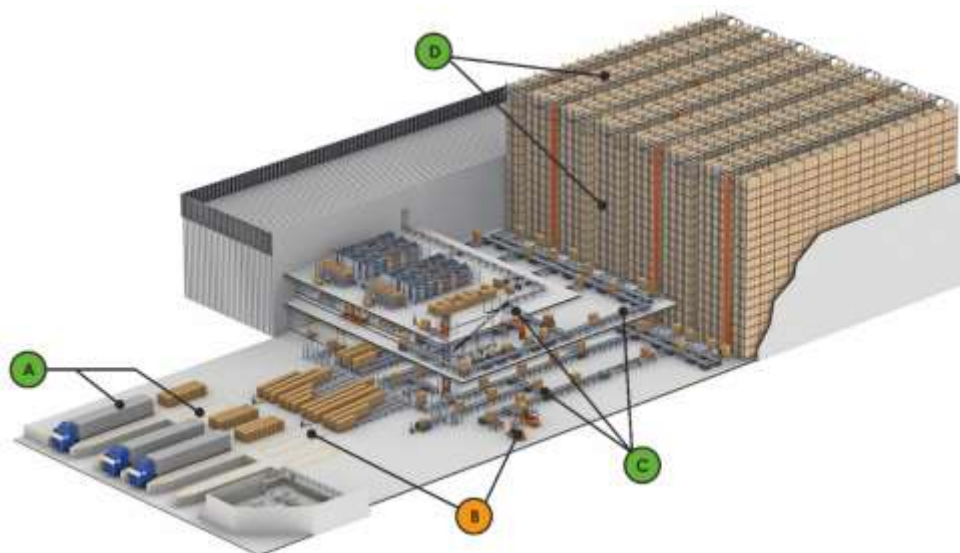


Рисунок 1.4 – Общая схема участка обработки и хранения грузов складского комплекса промышленного предприятия. А – приём/отгрузка; В – неавтоматизированная транспортировка с применением погрузчиков под управлением человека оператора; С, D – автоматизированная транспортировка. Источник изображения [8]

Фактически, до середины 2000-х годов, робототехнические решения применялись исключительно в зонах С и D (рисунок 1.4.) складских комплексов [9] и использовались для автоматизации:

- учёта объектов складирования;
- сортировки (роботы-сортировщики);
- паллетирования (роботы-паллетайзеры);
- стеллажного хранения (автоматические краны-штабелёры) [10].

Однако, в 2008 году с появлением на рынке АТМ (автономных транспортных модулей), ситуация изменилась кардинальным образом.

Так, например, Петр Кузнецов генеральный директор компании «Конфидент», одного из ведущих операторов рынка автоматизации складского хозяйства в России, характеризует текущую ситуацию следующим образом: «В складской логистике вообще происходят революционные изменения. IT-технологии принципиально изменили логистику. Мы строим роботизированные склады, где нет погрузчиков, все автоматизировано». [11] Домброва, Е. Роботы начали промышленную революцию в Петербурге // РБК Официальный сайт 08.01.2018 - Спб. [Электронный ресурс] – Режим доступа: https://www.rbc.ru/spb_sz/08/01/2018/5a5326ac9a794786d1d9d26f.

С появлением складских АТМ, началось новая ступень автоматизации складского хозяйства. Так, например, к 2025 году ожидается что более 50-ти тысяч складских комплексов полностью перейдут на использование АТМ, которые заменят классические способы транспортировки объектов хранения по территории склада (ричтраки [12]). Что вообще говоря не удивительно, так как экономический эффект от внедрения АТМ сложно переоценить.

Так, например, компания Амазон (Amazon), первопроходец в области применения АТМ для автоматизации складской логистики, заявляет о том, что с внедрением АТМ эффективность использования складских площадей в период с 2014 по 2017 увеличилась на 42%, а скорость обработки заказов выросла на 50%, что позволило компании перейти к системе доставки «День-в-день». Парк АТМ функционирующих на складах Амазон на 2017 год насчитывал 45000 единиц [13].

Компания Ди-Эйч-Эл (DHL), в отчёте о внедрении АТМ на тестовом складском комплексе за 2017 год, показала увеличение эффективности использования площадей складского комплекса на 60%, сокращении времени обработки грузов на 36%, а также увеличение производительности труда сотрудников в 2,5 раза. При этом в планы компании до 2022 года входит модернизация 60% складских комплексов по всей территории Северной Америки, путём внедрения АТМ. Это более 360 складских комплексов и логистических центров [13].

Британская компания Окадо (Ocado), сообщает о сокращении времени обработки грузов на 35% и увеличении эффективности использования складских площадей на 15% за счёт внедрения рельсовых АТМ разработанных собственным техническим подразделением компании [14].

Таким образом, несмотря на ряд технических ограничений, количество внедрений логистических АТМ неуклонно растёт по всему миру и по прогнозам аналитиков к 2021 году достигнет отметки в 486 тысяч единиц в год. Динамика роста приведена на рисунке 1.5 [13].

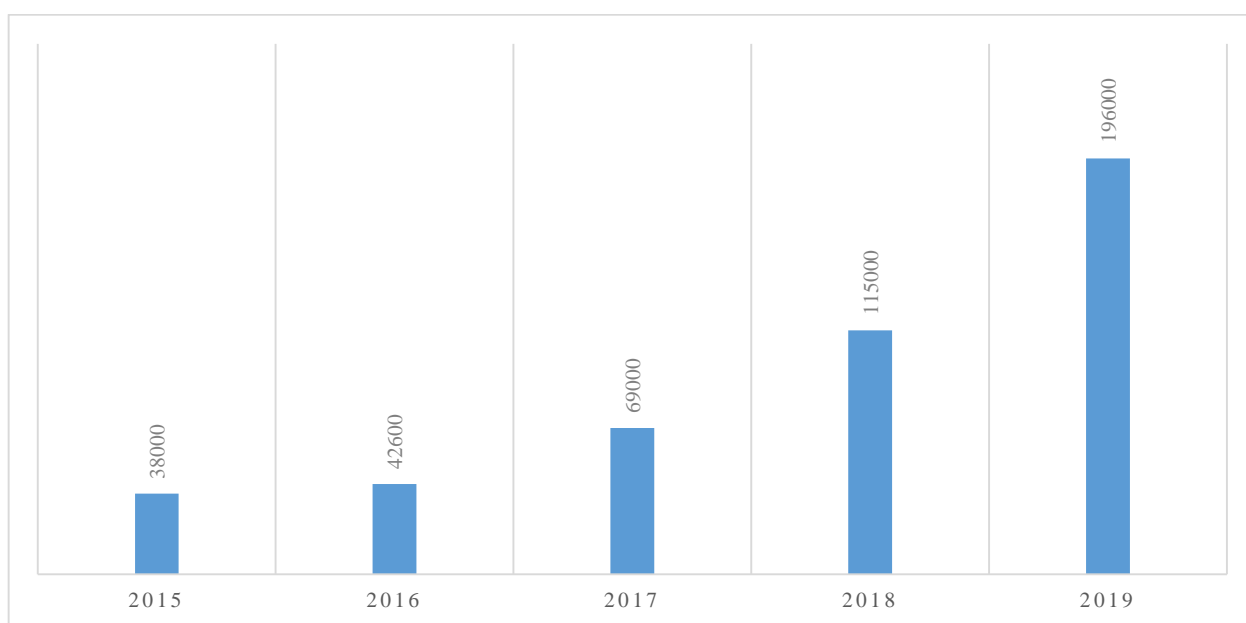


Рисунок 1.5 – Динамика роста поставок логистических АТМ в мире в период с 2015 по 2019 год. По материалам публикации [13]

1.2 Логистические АТМ, примеры внедрения, лидеры отрасли.

Лидирующие позиции на рынке логистических АТМ, занимают мобильные роботы транспортёры производства компании Амазон Роботикс (Amazon Robotics), в недавнем прошлом Кива Системс (Kiva Systems), пионера в разработке АТМ для складских комплексов, рисунок 1.6.



Рисунок 1.6 – АТМ производства Амазон Роботикс (Кива Системс) второе поколение, модуль грузоподъемностью 1360 кг. Источник изображения [15]

Руководители проекта Мик Маунтц (Mick Mountz), создатель Кива Системс, профессор Рафаэль Д'Андреа (Raffaello D'Andrea) и профессор Питер Вурманн (Peter R. Wurman).

Решение разрабатывалось по непосредственному заказу компании Амазон. Первые успешные испытания были проведены в 2008 году [16]. Демонстрация производилась на специально построенном для данной цели складском комплексе общей площадью 1000 м². Демонстрировались АТМ Кива Системс первого поколения, грузоподъемностью 450 кг рисунок 1.7.

Основная идея положенная в основу проекта робототехнического комплекса заключалась в том, чтобы не сотрудник перемещался по складу к стеллажу с объектами хранения, а напротив специально спроектированный стеллаж, так называемый под (Pod), доставляется к сотруднику по средствам АТМ. Таким образом сотрудник склада всегда остаётся статичным. Следует заметить, что доступ людей в рабочую зону АТМ запрещён.

Схема взаимодействия человек-робот характерная для решения Кива Системс приведена на рисунке 1.8. Ориентация АТМ в рабочей зоне осуществляется за счёт магнитной разметки напольного покрытия склада. Групповое взаимодействие и исключение коллизий обеспечивается посредством системы управления

верхнего уровня [16]. Для первого поколения АТМ стеллажи маркировались магнитными метками.



Рисунок 1.7 – АТМ производства Кива Системс модификация 2014 года, грузоподъёмность 450 кг. Источник изображения [16]

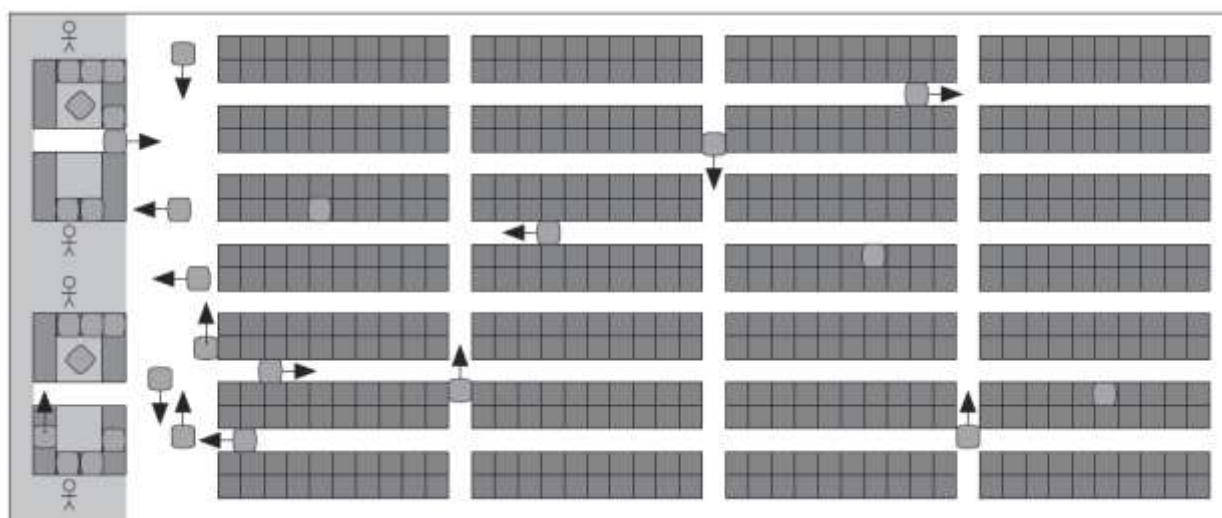


Рисунок 1.8 – Схема взаимодействия сотрудников склада и АТМ Кива Системс. Источник изображения [16]

Современные АТМ Амазон Роботикс выпускаются в двух модификациях, параметры которых приведены в таблице 1.1. [17]

Таблица 1.1 – Параметры АТМ Амазон Роботикс

Модель	Габариты, м			Грузоподъёмность, кг	Максимальная скорость, м/с
	Длина	Ширина	Высота		
Малые	1,25	0,99	0,15	450	1,3
Большие			0,2	1360	

Современные модификации АТМ Амазон Роботикс оснащены системой технического зрения, способной считывать штрих-коды, которыми маркируются стеллажи, имеют электрическую силовую установку, одной подзарядки батареи хватает на 90 минут непрерывной работы. Кинематическая схема четырёхколёсная с двумя ведущими колёсами и двумя роульными, свободно катящимися колёсными модулями. Тип маневрирования – силовой.

Прокладка маршрута и разрешение коллизий осуществляется системой управления верхнего уровня. Системы прокладки маршрута реализована по методу ориентированных графов [16].

Следует заметить, что на 2019 год у компании Амазон Роботикс, появилось конкуренты, в том числе среди отечественных производителей. На российском рынке аналогичное решение предлагает компания Ронави Роботикс (Ronavi Robotics), входящая в группу «ТехноСпарк» концерна РОСНАНО, рисунок 1.9.



Рисунок 1.9 – АТМ производства Ронави Роботикс. Исполнение N1500. Источник изображения [19]

Отечественное решение с технической точки зрения является полным аналогом АТМ Амазон Роботикс, существует в трёх исполнениях имеющих разную грузоподъёмность и назначение. Характеристики АТМ Ронави Роботикс приведены в таблице 1.2. [20].

Таблица 1.2 – Параметры АТМ производства Ронави Роботикс.

Модель	Габариты, м			Грузоподъёмность, кг	Максимальная скорость, м/с
	Длина	Ширина	Высота		
S50	0,75	0,6	0,4	500	1,5
M50					1,3
H1500	1,06	0,64	0,34	1500	

Технические решения, положенные в основу системы управления АТМ Ронави Роботикс, аналогичны техническим решениям Амазон Роботикс. Начало серийного производства АТМ Ронави Роботикс намечено на 2021 год.

Иной подход к созданию складских АТМ в 2016 году продемонстрировала компания Окадо (Ocado). Для автоматизации процесса транспортировки техническое подразделение компании под руководством Пола Кларка разработало робототехнический комплекс для собственных нужд, рисунок 1.10.

Перемещение роботов осуществляется по ортогональным рельсам, образующим секции, под которыми находятся складские ячейки. Управление АТМ, формирование маршрутов и разрешение коллизий осуществляется системой управления верхнего уровня. Роботы имеют четырёхколёсные пары, обеспечивающие движение во взаимно перпендикулярных направлениях. Роботы-транспортёры реализованы в виде куба со стороной 0,9 м. При перемещении развивают скорость до 1,4 м/с. Характер рабочей зоны исключает появление в ней человека [21]. В основе системы формирования маршрута лежит метод сеток предложенный Эльфом [22]. Взаимодействие между системой верхнего уровня и АТМ осуществляется посредством беспроводного канала связи с использованием оборудования стандарта 4G.



Рисунок 1.10 – Робототехнический комплекс компании Окадо. Источник иллюстрации [23]

Следует заметить, что система логистических АТМ разработанная компанией Окадо является узкопрофильной и не пригодна для внедрения в качестве средства автоматизации складов общего назначения, в том числе складов промышленных предприятий. О чём свидетельствует отсутствие внедрений системы сторонними по отношению к Окадо компаниями.

1.3 Перспективные решения в развитии логистических АТМ

Одним из наиболее перспективных решений в области разработки логистических АТМ, является проект ПЭН Роботс (PAN-Robots), который представляет собой совместную разработку целой группы крупных промышленных предприятий и университетов Западной Европы. В рабочую группу проекта входят мировой лидер в области производства промышленных датчиков внешней информации немецкая компания СИК (SICK), итальянская компания Элеттрик80 (Elettric80), группа компаний Касбега (Casbega), научно-исследовательские центры Модены и Финляндии.

Начиная с 2014 года проектной группой сделано более 60 публикаций по данной проблематике. Результатом исследований, проведённых рабочей группой проекта, является система ПЭН Роботс (PAN-Robots, сокращение от Plug&Navigate robots for smart factories). Это робототехнический комплекс, который позволяет реализовать перемещение объектов складирования по территории складского комплекса полностью исключая присутствие человека-оператора.

ПЭН Роботс имеет двухуровневую архитектуру [24], состоящую из системы управления верхнего уровня, интегрированной с помещением склада и нескольких АТМ (рисунок 1.11).

Система управления верхнего уровня включает в себя: центральную вычислительную систему, базу данных и массив распределённых по помещению складского комплекса потолочных лазерных сканеров, осуществляющих сбор информации об изменении геометрии пространства помещения склада. Архитектура системы проиллюстрирована в публикации [25]



Рисунок 1.11 – Структура системы ПЭН Роботс. Источник изображения [25]

Перемещение складированных объектов осуществляется АТМ с электрической силовой установкой, имеющие кинематическую схему трицикла [26] и способные развивать скорость до 1,3 м/с (рисунок 1.12). АТМ оснащены развитой системой датчиков внешней информации, позволяющей распознавать, классифицировать, определять ориентацию, скорость и направление движения окружающих транспортер объектов [28].



Рисунок 1.12 – ATM системы ПЭН Роботс. Использована иллюстрация из публикации [28]

Система датчиков внешней информации ATM является гибридной, состоящей из массива лазерных дальнометров и системы технического зрения реализованную по средствам 3D стереокамер. Реализация системы показана в публикации [30], снабжённых оптикой типа Фиш Ай (Fish-eye), подробно описанной в публикации [31]. Система лазерных дальнометров, взаимодействуя с распределёнными по территории складского комплекса маяками, обеспечивает возможность осуществлять определение собственной ориентации транспортёров в рабочей зоне (описание системы триангуляции положения ATM приведено в работе [32]).

Система технического зрения на базе оптики Фиш Ай разработана при участии научно-исследовательской группы компании SICK. Структура и пример реализации системы приведён в работах [33, 34]. Основной особенностью предложенной системы является возможность одновременного контроля всего пространства, окружающего ATM (360°). Ядром системы технического зрения является модуль распознавания образов, реализация которого описана в работе [35]. Предложенная система распознавания образов позволяет определять факт наличия объектов-пре-

пятствий на дистанции до 20м, классифицировать такие объекты, а также определять их ориентацию, направление и скорость движения относительно АТМ и своевременно принимать решение о корректировке маршрута движения, как это показано на рисунках 1.13 – 1.14.



Рисунок 1.13 – Поведение АТМ при обнаружении препятствия. Слева сверху – область обнаружения препятствий бортовой системой очувствления АТМ. Справа сверху, слева снизу – Вычисление отклонения от заранее запланированного маршрута с учётом пространственного расположения обнаруженных препятствий; Справа снизу – прерывание выполнения задания при невозможности уклониться от столкновения (защитная остановка). Источник [36]

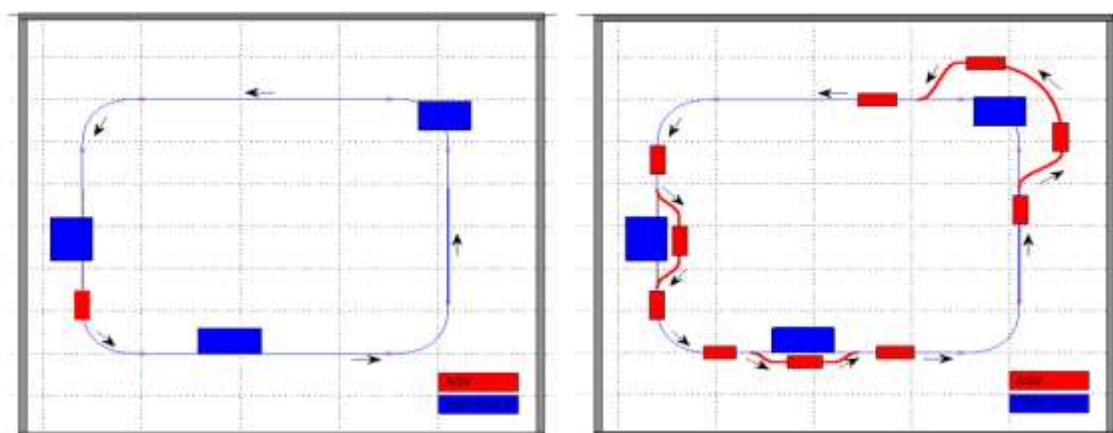


Рисунок 1.14 – График корректировки запланированного маршрута при обнаружении статического препятствия. Источник иллюстрации [36]

АТМ системы ПЭН Роботс является следующей ступенью эволюции логистических АТМ по сравнению с описанными выше АТМ Амазон Роботикс и им подобных реализаций. Так как, в данном случае система верхнего уровня отвечает исключительно за постановку задачи, то есть определение начальной и конечной точки перемещения, в то время как задача планирования маршрута, а также разрешение возможных коллизий возлагается на АТМ. Что с одной стороны существенно усложняет реализацию системы управления, с другой расширяет возможности и сферу применения данного решения, в том числе делает возможным присутствие человека в рабочей зоне АТМ.

Планирование перемещения АТМ осуществляется на основе комбинации карты местности, описанной в работе [37] и координатных сеток рабочего пространства представленной в работе [38]. Динамическая карта местности обновляется раз в 150мс, что позволяет избежать проблем характерных для условно детерминированных сред, а именно несоответствия расположения объектов нанесённых на карту местности их фактическому пространственному расположению, что показано в работе [28].

Формирование рабочей карты местности основано на трёхмерной статической карте рабочего пространства, записанной в систему на этапе внедрения. Препятствия, к которым относятся любые объекты, находящиеся в рабочей зоне (сотрудники склада или другие АТМ), наносятся на рабочую карту в режиме реального времени. Выявление препятствий осуществляется, во-первых стационарной подпотолочной системой слежения, во-вторых бортовой системой датчиков внешней информации. (описание функционирования динамической карты местности приведено в работе [24]).

Система управления «верхнего уровня» Джи-Эл-Ви (GLV, сокращение от Global Live View), осуществляет:

- распределение транспортных заданий между АТМ;
- управление взаимодействием между АТМ в рабочем пространстве;

- накопление и обработку данных о состоянии рабочего пространства;
- формирование динамической 3-х мерной карты рабочего пространства.

Обмен данными между системой управления верхнего уровня и АТМ, а также процесс формирования карты рабочего пространства представлен на рисунке 1.15. Иллюстрация подготовлена по материалам публикаций [28, 39].



Рисунок 1.15 – Генерации 3-х мерной карты рабочего пространства и координатных сеток складского помещения

На основании полученной 3-х мерной карты местности система ПЭН Роботс осуществляет формирование координатной сетки, описывающей текущее состояние рабочей зоны АТМ [39], сам процесс формирования координатной сетки описан в публикации [40], пример интерактивной координатной сетки рабочей зоны показан на рисунке 1.16.

Следует заметить, что в настоящий момент робототехнический комплекс ПЭН Роботс функционирует в демонстрационном режиме и находится в стадии контрольных испытаний.

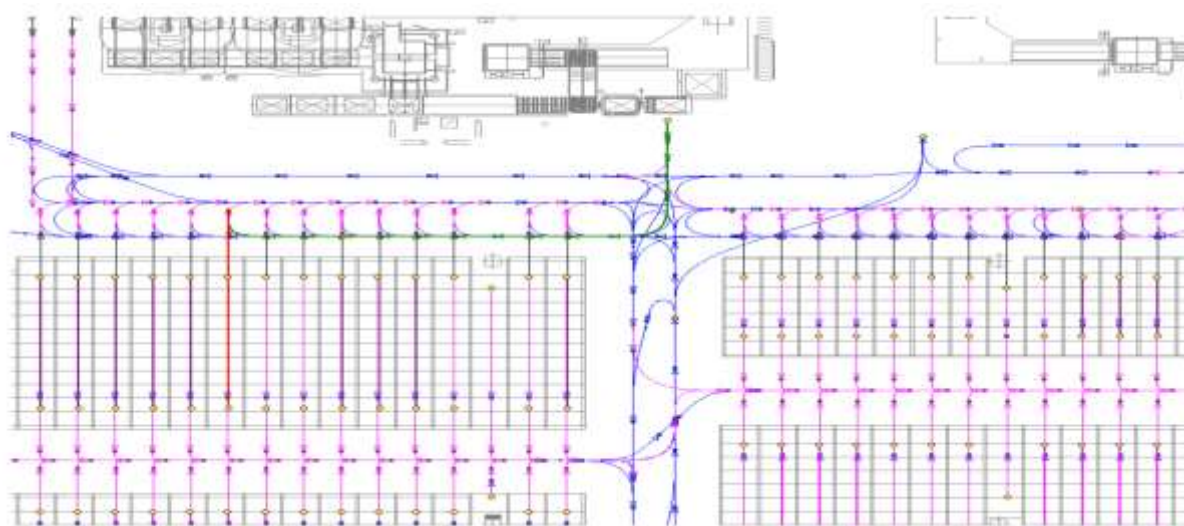


Рисунок 1.16 – Мгновенный срез координатной сетки рабочего пространства АТМ. Источник изображения [36]

Серийный выпуск системы намечен на 2030 год [41]. По мимо этого в 2019 году ещё несколько компаний анонсировало подобные ПЭН Роботс робототехнические комплексы для автоматизации складской логистики. Это китайская компания Ай-Кей-Ви Робот (IKV Robot). Эксплуатационные характеристики которого приведены в таблице 1.3, изображение аналога производства КНР приведено на рисунке 1.17 [42].



Рисунок 1.17 – АТМ логистического робототехнического комплекса производства компании Ай-Кей-Ви (КНР), аналог АТМ комплекса ПЭН Роботс. Источник иллюстрации [42]

Таблица 1.3 – Эксплуатационные характеристики АТМ Ай-Кей-Ви, приведённые производителем

Размеры АТМ, м	2,2 x 0,8 x 0,71
Тип датчиков	Система лазерных дальномеров
Канал связи	Wi-Fi
Управление направлением движения	Поворотные колесные модули
Напряжение питания	DC 24В
Грузоподъёмность	1000кг
Тип AGV	Вилочный погрузчик
Скорость движения	0,6 м/с
Радиус разворота, м	1,705-2,075 мм (зависит от комплектации)
Точность отработки расстояния, мм	± 10 мм
Перерегулирование по пространственному расположению	± 10 мм
Диапазон безопасности	Расстояние для экстренного торможения составляет менее 20 мм (регулируется)
Аккумулятор	Свинцово-кислотная батарея или литиевая батарея
Защитная защита	Датчик проверки переднего препятствия
	Механический механизм предотвращения столкновений Двойная защита
	Кнопка аварийного останова

Другим аналогом АТМ ПЕН Роботс, является АТМ производства Российской компании Робо-Си-Ви (RoboCV). Технические характеристики, российского аналога аналогичны АТМ производства Ай-Кей-Ви. Подробнее с проектом можно ознакомиться на официальном сайте компании [43]. Основной отличительной особенностью Российского аналога является то, что в качестве шасси используется штатный ричтрак на который устанавливается системы очувствления и автоматического управления движением. Иллюстрация Российского аналога приведена на рисунке 1.18.



Рисунок 1.18 – АТМ производства компании Робо Си-Ви на выставке в центра «Сколково». Источник иллюстрации [44]

1.4 Логистические АТМ. Рабочая зона, техническое оснащение, перемещаемые грузы

Одной из ключевых задач разработки перспективных логистических робототехнических комплексов, является делегирование задачи планирования маршрута

АТМ, что позволяет существенно упростить внедрение робототехнических комплексов на существующих складских предприятиях.

Выбор метода формирования маршрута АТМ предполагает наличие хотя бы базовой информации о рабочей зоне, в которой будет функционировать робот, конструкции шасси самого АТМ, его техническом оснащении, а также информации о типе перемещаемых грузов.

Складской комплекс. Будем исходить из того, что объектом автоматизации является складское помещение класса А+ по классификации Knight Frank. Склад представляет собой условно детерминированную среду, снабжённую вычислительной системой верхнего уровня, осуществляющей постановку задач для АТМ, а также системой контроля за перемещением объектов по территории рабочей зоны, аналогичной системе потолочных лазерных сканеров системы ПЭН Роботс [28]. Территория склада, на которой осуществляется транспортировка грузов, является крытой, защищённой от воздействий окружающей среды, и снабжена системой, поддерживающей постоянную температуру, соответствующую нормальным условиям. Транспортировка осуществляется на плоскости в пределах одного этажа. Поверхность перемещения, представляет собой ровный бетонный пол (или иное покрытие, предполагающее, что движение колёс погрузчика происходит без проскальзывания).

АТМ (автономный транспортный модуль). С целью классификации объекта управления в рамках настоящей диссертационной работы обратимся к ГОСТ 60.0.0.4-2018 «Роботы и робототехнические устройства. Термины и определения»:

- «2.6 робот (robot)*: Исполнительный механизм, программируемый по двум или более степеням подвижности (4.3), обладающий определенной степенью автономности (2.2) и способный перемещаться во внешней среде с целью выполнения задач по назначению.» [45] ГОСТ 60.0.0.4-2018, ст. 2.
- «2.13 мобильный робот (mobile robot): Робот (2.6), способный передвигаться под своим собственным управлением.» [45] там же, ст. 3.
- «3.16.1 колесный робот (wheeled robot): Мобильный робот (2.13), перемещающийся с помощью колес.» [45] там же, ст. 8.

Исходя из вышеуказанного объект управления может быть классифицирован как мобильный колёсный робот, так как:

- программируется по двум степеням подвижности;
- обладает автономностью, а именно способностью самостоятельно формировать маршрут;
- способен перемещаться в рабочем пространстве с целью выполнения задач по назначению;
- осуществляет перемещения под своим собственным управлением;
- перемещается по поверхности перемещения с помощью колёс.

АТМ реализован в виде четырёхколёсной мобильной платформы трансмиссия которой имеет четырёхколёсную автомобильную компоновку [46], то есть управляющая пара колёс представляет собой свободно катящиеся симметричные поворотные модули. Задние колёса выполнены в виде симметричных не поворотных колёсных модулей, являются ведущими. Колёсный робот снабжен системой датчиков внешней информации, аналогичной системе, применённой в проекте ПЭН Роботс и описанной в публикации [34], т.е. в любой момент времени системе управления доступны данные об окружающих его препятствиях, их классе, геометрии, ориентации в пространстве, скорости и направлении движения. Помимо этого, бортовая вычислительная система АТМ в любой момент времени имеет информацию о собственном пространственном расположении относительно глобальной системы координат склада (реализация разметки рабочей зоны транспортёра аналогична, предложенной в работе [47]). Вариант возможной реализации системы триангуляции положения транспортёра в пространстве рабочей зоны показан в публикации [39]. При движении скорость колёсного робота не превышает 1,5м/с [48], колёса транспортёра движутся по поверхности перемещения без проскальзывания. Мобильная платформа имеет электрическую силовую установку и оснащена исполнительной системой, способной принимать цифровой сигнал и преобразовывать его в угол поворота управляющих направлением движения колёс. Система управ-

ления скоростью движения АТМ, способная поддерживать скорость движения. Помимо мобильной платформы в состав АТМ входит устройство, позволяющее осуществлять загрузку и разгрузку транспортёра в автоматическом режиме.

Система датчиков внешней информации. Предполагается, что система очувствления робототехнического комплекса имеет двухуровневую реализацию, схема которой аналогична применённой в проекте ПЭН Роботс.

Система управления верхнего уровня выполняет функции объективного контроля пространственного расположения АТМ и диспетчера транспортных задач. Оснащена потолочными лазерными сканерами, осуществляющими сбор информации об изменении геометрии рабочего пространства в котором функционируют АТМ.

В качестве потолочных сканеров могут быть использованы сканеры Фаро Фокус S350 (FARO FOCUS S350), производства компании ФАРО (FARO). Технические характеристики сканеров приведены в официальной документации [49]. Программный интерфейс (API) сканеров может быть реализовано в виде программного модуля Фаро Сцен (FARO SCENE) [50], обеспечивающего системную интеграцию сканеров в инфраструктуру робототехнического комплекса, а также предварительную обработку полученных данных.

На основании полученных данных сканеров формируется карта рабочей зоны. Процесс создания карты рабочей зоны описан в публикации [51]. Конечная реализация может быть сделана посредством программного пакета Орджин (Origin) [52], имеющего необходимый набор инструментов, позволяющий работать в фоновом режиме, и имеющий хорошо задокументированный программный интерфейс.

Система очувствления АТМ, является гибридной и состоит из массива лазерных дальномеров, формирующих массив данных о дистанциях до потенциальных препятствий. В качестве которых могут быть использованы дальнометры Скансе Свип ЛИДАР (Scanse Sweep LIDAR), производства компании Скансе (Scanse), технические характеристики прибора [53], описание API и руководство

по системной интеграции [55]. Методика получения и интерпретации данных с примером реализации показана в публикациях [55, 56].

Коррекция полученных данных о препятствиях окружающих АТМ осуществляется посредством стереокамер, снабжённых оптикой Фиш Ай разработана компанией Зимакс (Zemax) при участии научно-исследовательской группы компании Сик (SICK) [57]. В качестве стереокамер в проекте могут быть применены камеры 3D-A5000, производства Когнекс (Cognex) на базе программного обеспечения ВиженПро (VisionPro) [58]. Реализация системы распознавания образов и вычисления коллизий может быть выполнена посредством программного обеспечения Зимакс ОптикСтудио (Zemax OpticStudio) [57].

Перемещаемые объекты. Предполагается, что объекты складирования представляют собой материалы, конструкционные элементы и объекты готовой продукции машиностроительных предприятий, не требующие особых условий транспортировки. То есть они не относятся к классам прецизионных механизмов, легковоспламеняющихся, взрывчатых веществ или боеприпасов. Таким образом, в рамках работы считается, что ограничения, связанные с линейными и боковыми ускорениями при перемещении грузов, отсутствуют. Также считается, что при перемещении груз находится на паллете, габариты груза по длине и ширине не превышают размеров паллеты. Масса груза не превышает 2000 кг.

1.5 Краткая классификация методов планирования маршрута АТМ

К настоящему моменту в литературе описано огромное множество методов планирования маршрута АТМ, которые реализуют различные подходы, являющиеся следствием, конкретного применения АТМ. Фактически каждая отдельная группа методов планирования маршрута мобильного робота является отдельной темой для самостоятельного исследования.

Данная задача лежит далеко за рамками настоящей работы, поэтому ограничимся краткой классификацией наиболее распространённых методов планирования маршрута АТМ.

На рисунке 1.19 приведена классификация наиболее часто встречающихся групп методов планирования маршрута для АТМ.



Рисунок 1.19 – Классификация методов планирования маршрута АТМ [59].

Следует заметить, что результатом подавляющего большинства методов является последовательность опорных точек, определяющих пространственное расположение АТМ на маршруте, что приводит к необходимости сглаживания маршрута.

Методы на основе графов. Широко распространённые методы данного типа показаны на рисунке 1.20.

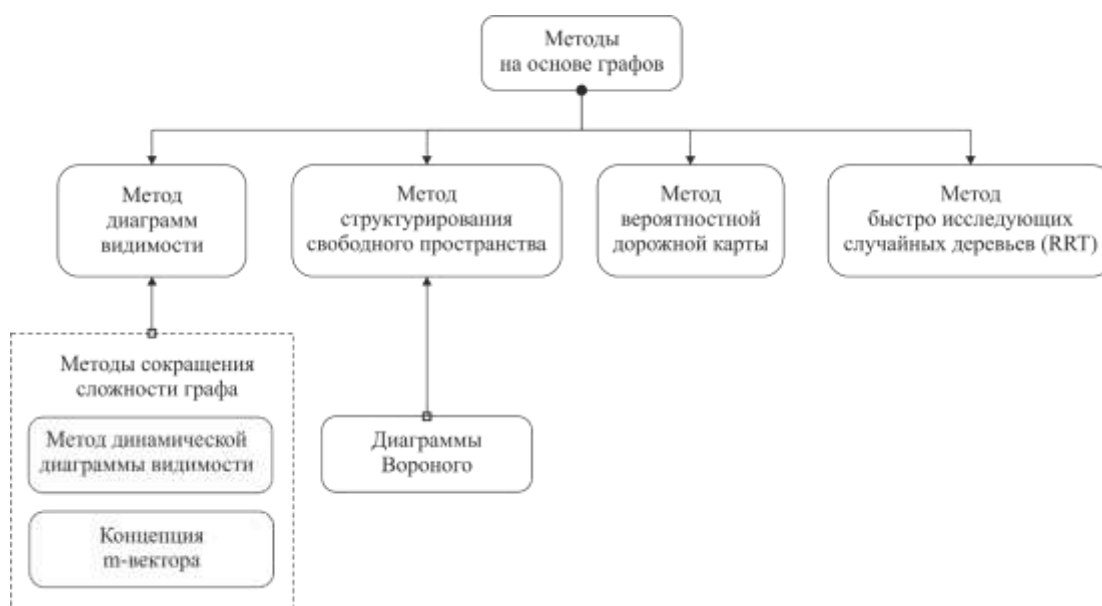


Рисунок 1.20 – Методы планирования маршрута АТМ на основе графов. Иллюстрация выполнена по материалам [59]

Методы планирования маршрута на основе графов подразделяются на методы диаграмм видимости [60, 61], методы структурирования свободного пространства (и их разновидность - диаграммы Воронова) [62, 63], метод вероятностной дорожной карты [64] и методы быстро исследующих случайных деревьев (RRT методы) [65].

Методы на основе графов характерны для планирования маршрутов в детерминированных средах (в частности они используются для планирования движения логистических АТМ Амазон Роботикс).

Методы на основе клеточной декомпозиции. В основе данной группы методов лежит идея дискретизации рабочей зоны АТМ. Классификация методов на основе клеточной декомпозиции приведена на рисунке 1.21.

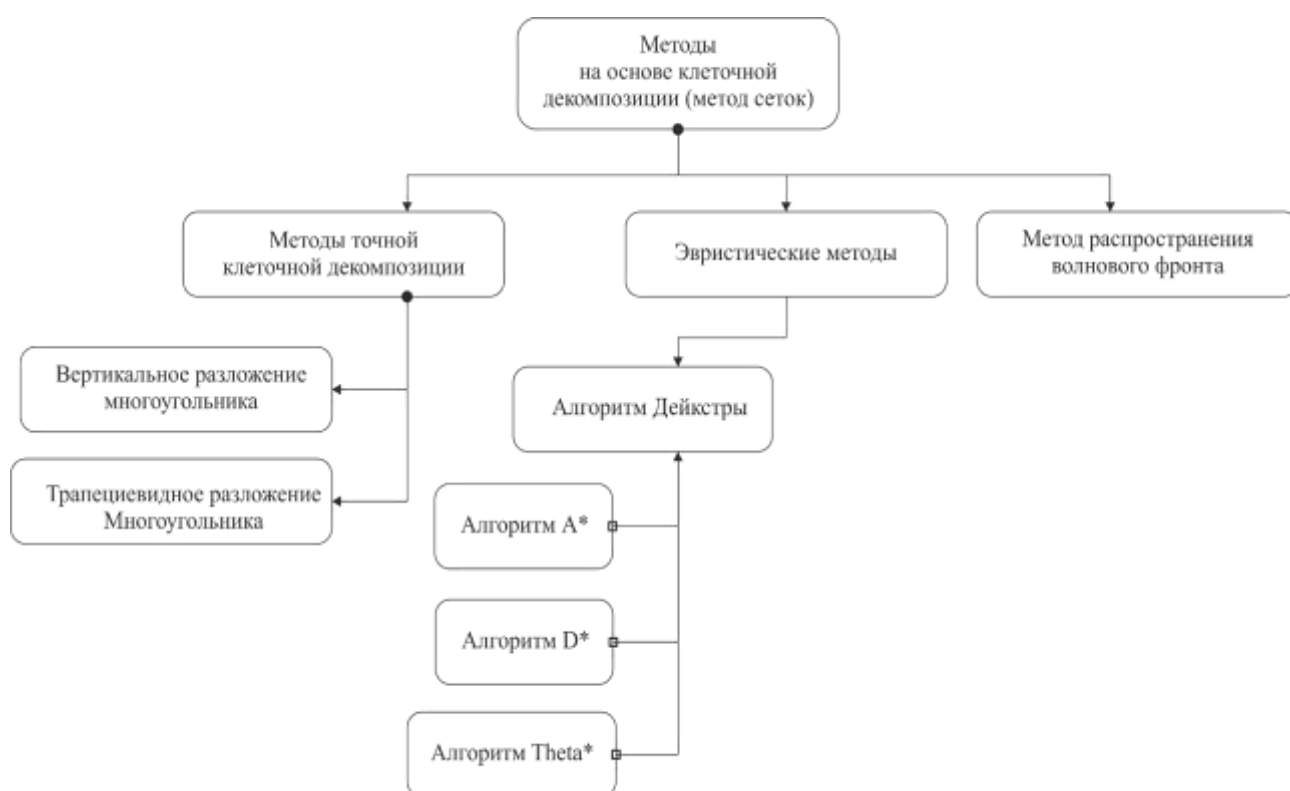


Рисунок 1.21 – Методы планирования маршрута АТМ на основе клеточной декомпозиции. Иллюстрация выполнена по материалам [59]

К методам клеточной декомпозиции относятся: точная клеточная декомпозиции [66], приближенная клеточная декомпозиция (метод сеток) [66], ряд эвристических методов, например, алгоритм Дейкстры (и его

вариации A^* алгоритм [67], D^* алгоритм [68] и Θ^* [69]), а также метод рас-пространения волнового фронта [70]. Заметим, что метод сеток использо-ван для реализации планирования маршрутов логистических АТМ компании Окадо.

Методы потенциальных полей. Общая идея данной группы методов заклю-чается в том, что АТМ перемещается вдоль векторных линий векторного поля, по-тенциальная функция которого отражает форму препятствий и заданное простран-ственное расположение. Классификация методов потенциальных полей приведена на рисунке 1.22.

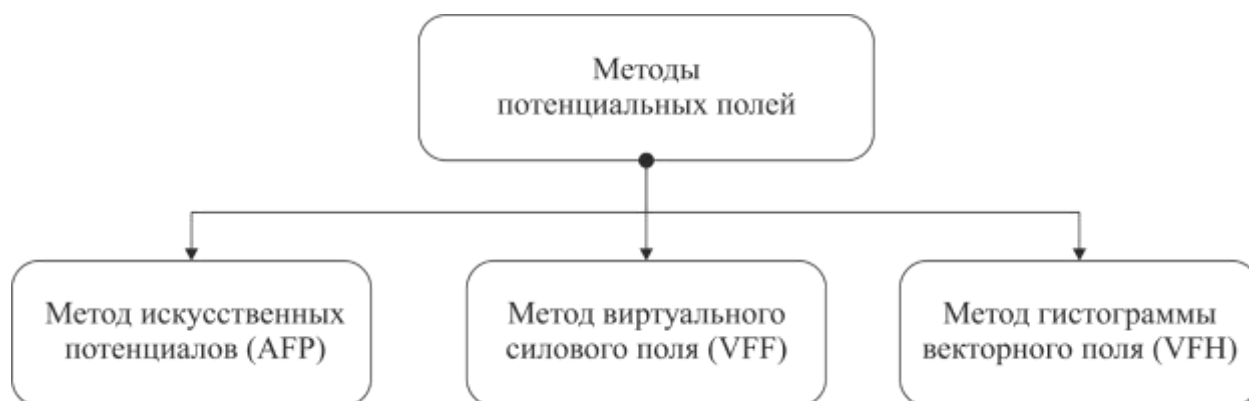


Рисунок 1.22 – Классификация методов планирования маршрута АТМ на основе потенциальных полей. Иллюстрация выполнена по материалам [59]

К данной группе методов относятся метод искусственных потенциалов [71], метод виртуального силового поля [72], а также метод гистограммы векторного поля [73]. Несмотря на то, что данная группа методов является одной и самых популярных при планировании локального пути АТМ, тем не менее все методы данной группы имеют существенный недостаток, а именно возможное существование локальных минимумов. И несмотря на то, что более поздних вариациях [72, 73] предпринята попытка устранения данного недостатка полностью удовлетворительного решения пока не найдено [59].

Оптимизационные методы. Задача планирования пути АТМ может решаться как задача оптимального управления. При этом движение АТМ нужно пред-ставить в виде динамической системы. Препятствия описываются как некоторые

ограничения, а допустимая траектория оценивается некоторой целевой функцией. Как правило при планировании маршрута АТМ прямая задача оптимального управления сводится к дискретной задаче нелинейного программирования. Классификация оптимизационных методов приведена на рисунке 1.23.



Рисунок 1.23 – Оптимизационные методы планирования маршрута АТМ. Иллюстрация выполнена по материалам [59]

К наиболее распространённым подходам к планированию маршрута АТМ как к задаче оптимального управления относятся: плоские модели [74], смешанное целочисленное линейное программирование [75, 76], а также двоично-целочисленное программирование [77]. Данная группа алгоритмов планирования маршрута АТМ является одной из наиболее универсальных и подходит для решения самого широкого круга задач. Однако при этом реализация оптимизационных методов «в железе», является крайне трудоёмкой задачей [59].

Интеллектуальные алгоритмы. Это алгоритмы имитирующие процессы, протекающие в живой природе, поведение биологических существ, мышление человека или поведение биологических сообществ. В действительности к данной группе относится огромное множество алгоритмов, описанных в литературе. На рисунке 1.24 приведём только те которые получили широкое распространение в мобильной робототехнике.



Рисунок 1.24 – Интеллектуальные алгоритмы планирования маршрута АТМ. Иллюстрация выполнена по материалам [59]

К данной группе алгоритмов планирования маршрута для АТМ относятся:

- муравьиный алгоритм [78], который как правило применяется для планирования маршрута на основе карты или графа;
- искусственная нейронная сеть [79, 80], которая используется в задачах адаптивного управления движением АТМ;
- метод роя частиц (МРЧ) [81], характерный для формирования маршрутов беспилотных летательных аппаратов;
- реактивные методы [63], имитирующий поведенческие реакции живых организмов.

Подводя итог под приведённой классификацией, следует сказать, что методы на основе графов имеют простую машинную реализацию, однако хорошо работают только в детерминированных средах, хотя некоторые из них могут быть модифицированы для использования в средах с динамическими препятствиями. Так как граф строится по опорным точкам, то полученный маршрут всегда нуждается в последующем сглаживании.

Основным преимуществом группы методов на основе потенциальных полей является простота реализации и малое время расчёта, что послужило причиной его широкого распространения в задачах планирования локальных маршрутов АТМ, однако к его недостаткам следует отнести наличие локальных минимумов, в которые может попасть АТМ в конечном счёте не достигнув заданного пространственного расположения.

Оптимизационные методы, позволяют решать задачи с учётом самых сложных ограничений, при этом позволяют вводить дополнительные ограничения направленные, например, на обеспечение безопасности движения. и в принципе являются самым универсальным инструментом планирования маршрутов для АТМ, но при этом мало пригодны для использования в системах реального времени так как в большинстве случаев требуют качественно иного уровня (по сравнению с прочими методами) детализации модели, что приводит к замедлению процесса вычисления маршрута.

Методы, основанные на интеллектуальных алгоритмах, как и оптимизационные алгоритмы позволяют решать задачи со сложными ограничениями, но зачастую имеют слишком длительный цикл планирования. Такие методы лучше всего применять при первоначальном глобальном планировании маршрутов.

1.6 Интеграция нескольких методов планирования маршрута АТМ

Как правило, современные АТМ используют комбинированные алгоритмы планирования маршрута, к примеру, логистические АТМ ПЭН Роботс, одновременно используют карту местности, искусственную нейронную сеть и алгоритм типа Баг (Bug) [83], АТМ Окадо одновременно применяют метод координатных сеток и метод роя частиц.

Такой подход к планированию маршрута позволяет интегрировать преимущества различных методов и нивелировать их недостатки. Так, например, сочетание локальных и глобальных методов планирования, позволяет планировать глобальный маршрут на основании конфигурации рабочей зоны, которая может быть постоянной, а локальные метод позволят АТМ маневрировать на при выполнении задания с целью ухода от столкновений с подвижными препятствиями.

Большинство методов планирования маршрута в приведённой выше классификации относится к методам глобального планирования (за исключением метода

потенциальных полей). Рассмотрим пример альтернативного широко распространённого метода локального планирования маршрута АТМ.

Bug-алгоритм — один из самых первых и широко распространённых алгоритмов разрешения коллизии АТМ и препятствия. Основная идея, положенная в основу алгоритма, заключается в следующем: от текущего пространственного расположения к заданному пространственному расположению АТМ прокладывает прямой маршрут и движется по нему до тех пор пока не «сталкивается» с препятствием.

На вход системы реализующей Bug-алгоритм поступает только информация о состоянии пространства непосредственно окружающего АТМ, а также информация о заданном пространственном расположении. Алгоритм реализует два режима движения: «движение к заданному пространственному расположению» и «обход препятствий». Вариации алгоритма различаются способом реализации обхода препятствия. На рисунке 1.25 показаны 3 различные реализации Bug-алгоритма.

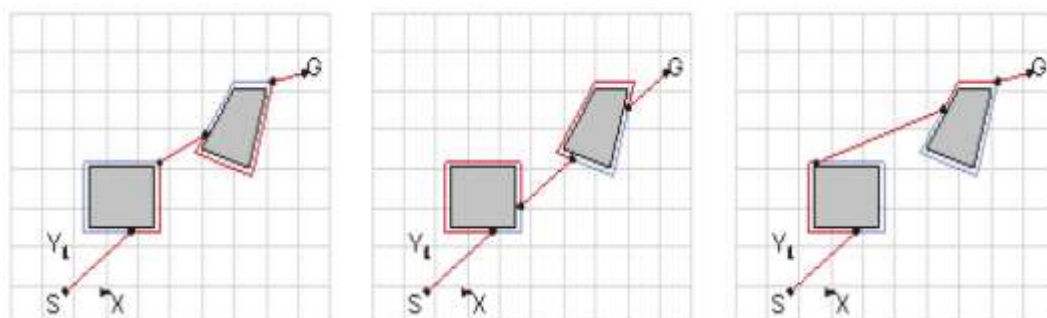


Рисунок 1.25 – Реализации Bug алгоритма. Слева Bug-1; По центру Bug-2; С права DistBug алгоритм; S – исходное пространственное расположение; G – заданное пространственное расположение [59]

Алгоритм Bug-1 [83]:

1. Движение в направлении заданного пространственного расположения;
2. Если встречено препятствие, происходит огибание его до тех пор, пока АТМ не выйдет в точку, ближайшую к заданному пространственному расположению;
3. Продолжение движения к заданному пространственному расположению.

Алгоритм Bug-2 [84]:

1. Движение в направлении заданного пространственного расположения, фиксируется наклон прямой маршрута;
2. Если встречено препятствие, происходит огибание его до тех пор, пока текущий наклон прямой маршрута не совпадет с зафиксированным наклоном;
3. Продолжение движения к заданному пространственному расположению.

С правой стороны рисунка 1.25 показан алгоритм ДистБаг (Distbug) [84], в котором с целью определения момента начала уклонения от столкновения с препятствием используется массив данных в котором хранится ранее накопленная информация о предыдущих опытах маневрирования.

На приведённых алгоритмов Bug 1 и Bug 2 разработана масса вариаций Bug-алгоритма [83, 84]. Bug-алгоритмы относятся к классу реактивных и используются для перемещения АТМ в недетерминированных или условно детерминированных средах, которые плохо поддаются формализации.

1.7 Выводы по главе 1

1. Разработка логистических АТМ является актуальной научно-технической проблемой, что подтверждается динамикой спроса на подобные решения на рынке робототехники.

2. Наиболее перспективным типом логистических АТМ, являются модули способные адаптироваться к изменениям в рабочей зоне и самостоятельному планированию маршрута движения в пределах рабочей зоны, так как это существенно расширяет возможности последующего внедрения таких модулей на существующих складских площадях.

3. Одной из ключевых задач разработки перспективных логистических АТМ является разработка алгоритмов локального планирования маршрута. Основной целью таких алгоритмов является уклонение АТМ от столкновения с препятствием.

Глава 2. Кинематическое управление движением четырёхколёсных АТМ

2.1 Скоростные режимы движения логистических АТМ

Одним из важнейших параметров при проектировании любого АТМ, является максимально допустимая скорость движения робота внутри рабочей зоны. Так как именно от желаемых скоростных характеристик во многом зависит выбор конструкции шасси АТМ, а также выбор технических решений при разработке системы планирования маршрута и выполнения задания.

Следует заметить, что в случае логистических АТМ максимально допустимая скорость движения в рабочей зоне зависит не столько от технических возможностей мобильной платформы, сколько от нормативно правовой базы, регламентирующей скоростные режимы на территории складских комплексов. Так если рассматривать существующие модели логистических АТМ, то можно видеть, что максимальные скорости движения заложенные разными производителями примерно одинаковы, что проиллюстрировано на рисунке 2.1.

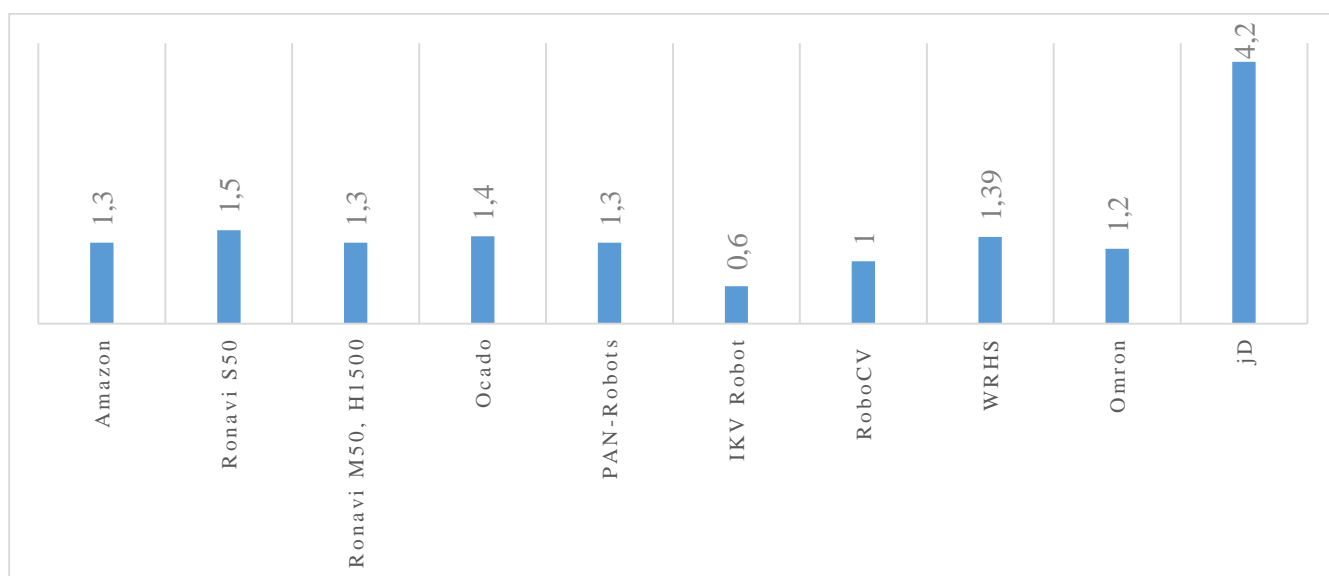


Рисунок 2.1 – Максимальные скорости развиваемые логистическими АТМ разных производителей

Что в первую очередь обусловлено международными правилами эксплуатации погрузчиков на территории складов и логистических центров. В Российской Федерации документом, регламентирующим скорость движения погрузчиков по территории складского комплекса является «ТИПОВАЯ ИНСТРУКЦИЯ ПО ОХРАНЕ ТРУДА ДЛЯ ВОДИТЕЛЕЙ АВТОПОГРУЗЧИКОВ. ТИ Р М-009-2000" (утв. Минтрудом РФ 17.03.2000)» [85]. В соответствии с пп. 1.20, которой предельно допустимая скорость движения погрузчика по территории крытого склада составляет 5км/ч, что соответствует 1,38 м/с.

Так как объект автоматизации с юридической точки зрения классифицируется как «склад», то по крайней мере до внесения изменений в нормативно-правовую базу, разработка более скоростных моделей логистических АТМ является не целесообразной.

Что с одной стороны (экономической) является негативным фактором, определяющим «потолок» пропускной способности складского предприятия. С другой стороны (технической), данные скоростные ограничения позволяют упростить процесс проектирования логистических АТМ. Так как при движении с малыми скоростями можно без потери практической ценности полученного результата применять простые кинематические модели при проектировании системы управления движением АТС.

2.2 Рулевое управление Аккермана. Кинематика поворота четырёхколёсного транспортного средства.

Транспортные средства имеющих автомобильную кинематическую схему, в независимости от функционального назначения, для маневрирования в подавляющем большинстве случаев используют рулевое управление, конструкция которого основана на принципе Аккермана, подробно описанном в классических работах посвящённых конструкции четырёхколёсных транспортных средств Е.А. Чудакова,

Й. Раймпеля, П.В. Аксенова, Д.А. Антонова, А. Дика, Р.П. Кушвида, В.Ф. Платонова, Д.Р. Эллиса, Е.О. Рыкова, Ю.В. Шемякина, Г.П. Антипова, Н.А. Алфутова, Я.Е. Фаробина и др.

Рулевое управление Аккермана [86, 87] (в ряде классических источников Жанто [88, 89, 90, 91]), имеет конструкцию, показанную на рисунке 2.2.

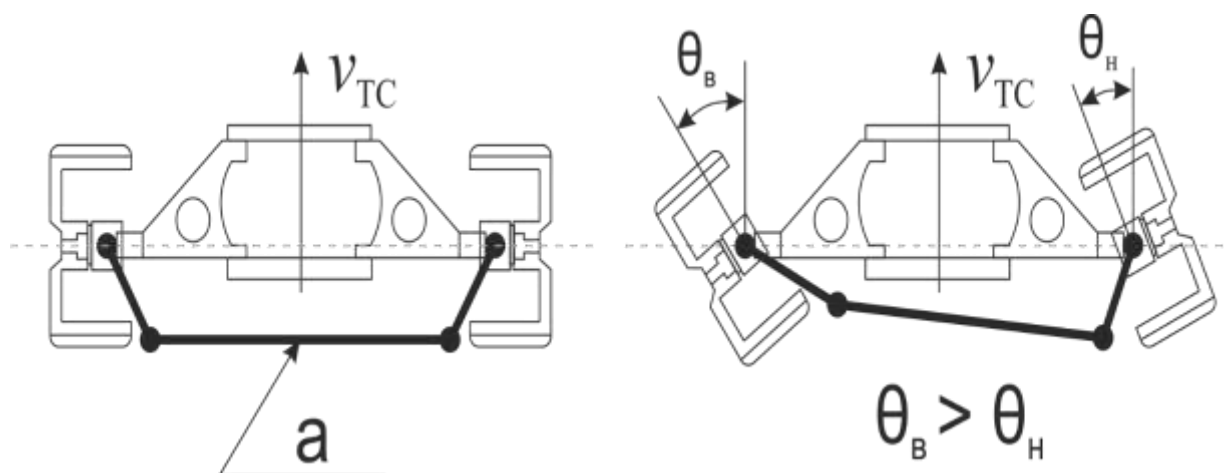


Рисунок 2.2 – Конструкция рулевого управления Аккермана (Жанто). Слева – колёса прямо; Справа – положение колёс ТС во время поворота; а – схема установки рулевых рычагов реализующих принцип Аккермана; $v_{ТС}$ – направление движения ТС; $\theta_{в}$, $\theta_{н}$ – углы отклонения от продольной оси ТС внутреннего и наружного колеса во время выполнения поворота. Источник иллюстрации [92]

Принцип Аккермана заключается в неравномерности углового перемещения управляющих колёс (рисунок 2.2), что обусловлено установкой рулевого рычага под углом к продольной оси ТС. Причём, угловое рассогласование между управляющими колёсами является нарастающим.

Шарнир В рулевого рычага (рисунок 2.3) находится на окружности, которую будет описывать точка В при повороте управляемого колеса.

Рассогласование углов отклонения управляющих колес позволяет получить параллельное расположение следов колес как при прямолинейном движении ТС, так и при выполнении поворота.

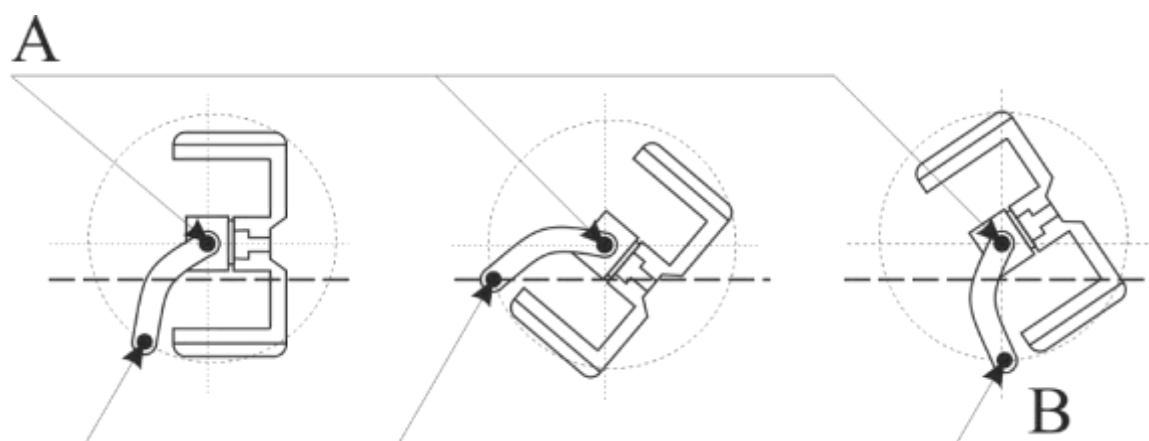


Рисунок 2.3 – Перемещение рулевого рычага при вращении рулевого колеса. А – центр вращения управляющего колеса ТС относительно вертикальной оси; В – точка крепления рулевого рычага к рулевой рейке. Источник иллюстрации [92]

Таким образом при точном угле Аккермана плоскости вращений обоих передних колес принадлежат касательным, построенным в центральных точках пятен контакта передних управляемых колес с поверхностью перемещения [88, 89, 91].

Рассмотрим поворот ТС без учета бокового увода. Будем считать, что боковое скольжение колёс при выполнении поворота отсутствует. В этом случае векторы скоростей центров всех колес \vec{v}_{ki} должны лежать в плоскостях их вращения. Следовательно, управляемые колеса необходимо повернуть на такие углы, при которых перпендикуляры к плоскостям вращения колес пересекутся в одной точке — мгновенном центре скоростей. Эту точку называют мгновенным центром поворота.

На рисунке. 2.4 показана схема поворота ТС при качении колес без бокового увода.

Мгновенный центр поворота находится в точке O на продолжении оси неуправляемых колес. Радиусом поворота ТС – R_{Π} называется расстояние от мгновенного центра поворота O до продольной оси ТС обозначенной AB .

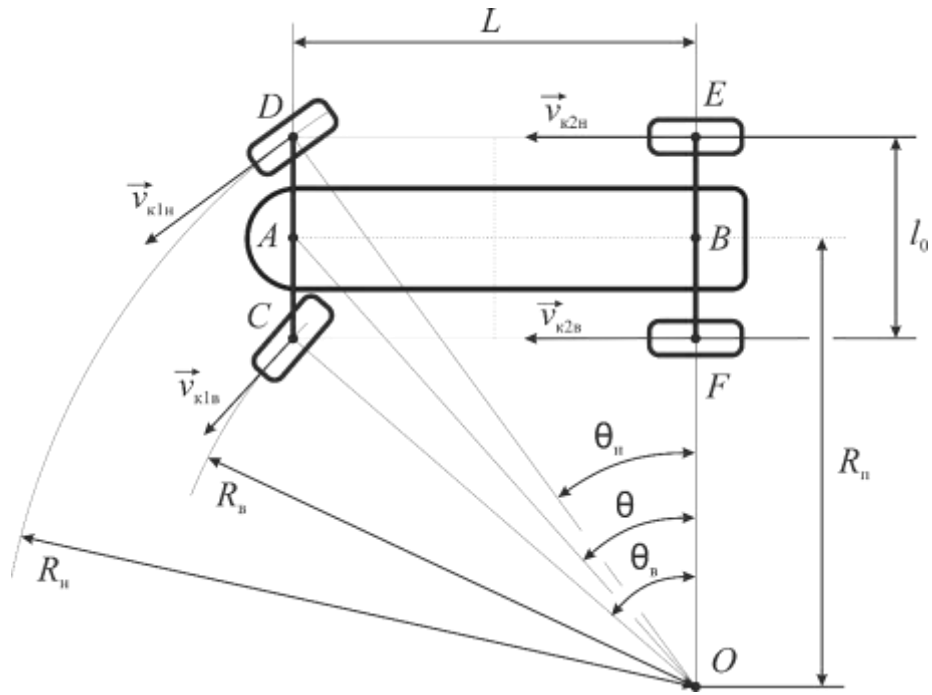


Рисунок 2.4 – Схема поворота ТС при качении колёс без бокового увода. Источник иллюстрации [46]

Найдем соотношение между углами поворотов наружного θ_n и внутреннего θ_b управляемых колёс. Из треугольников ODE и OCF :

$$\operatorname{ctg}\theta_n = \frac{OE}{DE}; \operatorname{ctg}\theta_b = \frac{OF}{CF}. \quad (2.1)$$

Вычитая из первого равенства второе и принимая во внимание, что

$$DE = CF = L, OE - OF = l_0,$$

получаем:

$$\operatorname{ctg}\theta_n - \operatorname{ctg}\theta_b = \frac{l_0}{L}, \quad (2.2)$$

где l_0 – расстояние между осями шкворней поворотных цапф управляемых колёс ТС.

При изменении углов поворота в достаточно широких пределах (для современных ТС максимальные углы поворота колес достигают $35 - 46^\circ$) классическая конструкция рулевой трапеции Аккермана (Жанто) [86] (см. рисунок 2.2) не обеспечивает точного выполнения условия Аккермана (равенство 2.2), что компенсируется за счёт:

- изменения угла наклона рычагов управляющих колёс по отношению к продольной оси ТС;
- боковой деформации шин, а также их бокового скольжением при выполнении поворота [88, 89, 92].

Таким образом, при проектировании современных ТС углы наклона поворотных рычагов выбирают таким образом, чтобы в положении рулевой трапеции, соответствующем движению ТС по прямой, осевые линии рычагов пересекались в некоторой точке E расположенной внутри корпуса ТС, как это показано на рисунке 2.5.

При этом точка E должна находиться на расстоянии l_E от оси вращения управляемых колес, определяемым соотношением

$$l_E = k_E L,$$

где, значение коэффициента k_E находится в пределах $0,7 - 0,9$. [46].

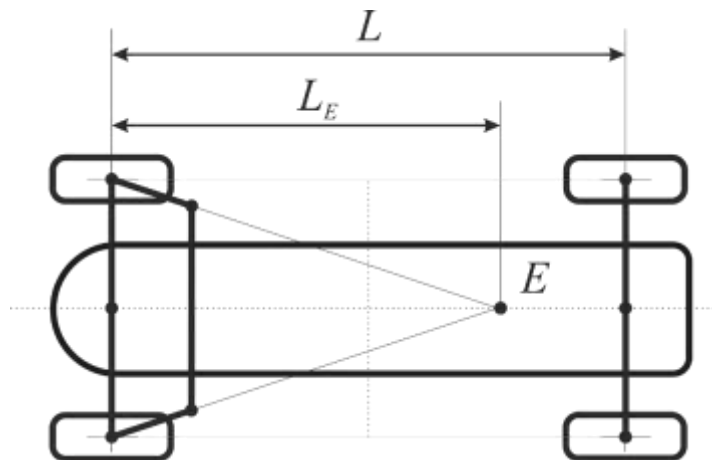


Рисунок 2.5 – Схема выбора параметров рулевой трапеции [46]

То есть в процессе проектирования системы рулевого управления современных ТС осуществляется поиск оптимального значения k_E , при котором обеспечивается наилучшее согласование углов поворота управляемых колес поворота θ_H и θ_B .

Для определения зависимости R_{Π} от углов поворота управляемых колес сложим левые и правые части выражений (2.1). Принимая во внимание, что

$$OE = R_{\Pi} + \frac{l_0}{2}; OF = R_{\Pi} - \frac{l_0}{2},$$

получаем:

$$ctg\theta_B + ctg\theta_H = \frac{2R_{\Pi}}{L}. \quad (2.3)$$

Из рисунка 2.4 следует, что [46]:

$$ctg\theta = \frac{ctg\theta_H + ctg\theta_B}{2}, \quad (2.4)$$

где θ – средний угол поворота управляемых колёс, при сравнительно небольших углах поворота управляемых колес, характерных для обычных эксплуатационных условий (*кроме маневрирования на площадках ограниченной величины*), в первом приближении можно принять [46]:

$$\theta \approx \frac{\theta_H + \theta_B}{2}.$$

Тогда

$$R_{\Pi} = \frac{L}{tg\theta}. \quad (2.5)$$

При движении ТС на повороте с высокими скоростями возникают значительные боковые составляющие сил инерции, поэтому колеса катятся с боковыми уводами. В общем случае углы увода всех колес различны. Однако с целью упрощения

во многих случаях без существенной ошибки можно принимать для колес одного и того же моста одинаковые углы увода [46]

$$\delta_{yvi} = \frac{\delta_{yв.нi} + \delta_{yв.вi}}{2}.$$

В этом случае угол δ_{yvi} называют углом увода i -ого моста.

При возникновении бокового увода векторы скоростей центров колес \vec{v}_{ki} не лежат в плоскостях их вращения, а отклоняются в сторону действия боковых сил на углы увода $\delta_{yв1}$ и $\delta_{yв2}$ как показано на рисунке 2.6.

Вследствие этого мгновенный центр поворота перемещается из точки O в точку O^* , а радиус поворота $R_{п}^*$ становится равным расстоянию O^*D , где O^*D — длина перпендикуляра, опущенного из точки O^* на продольную ось ТС.

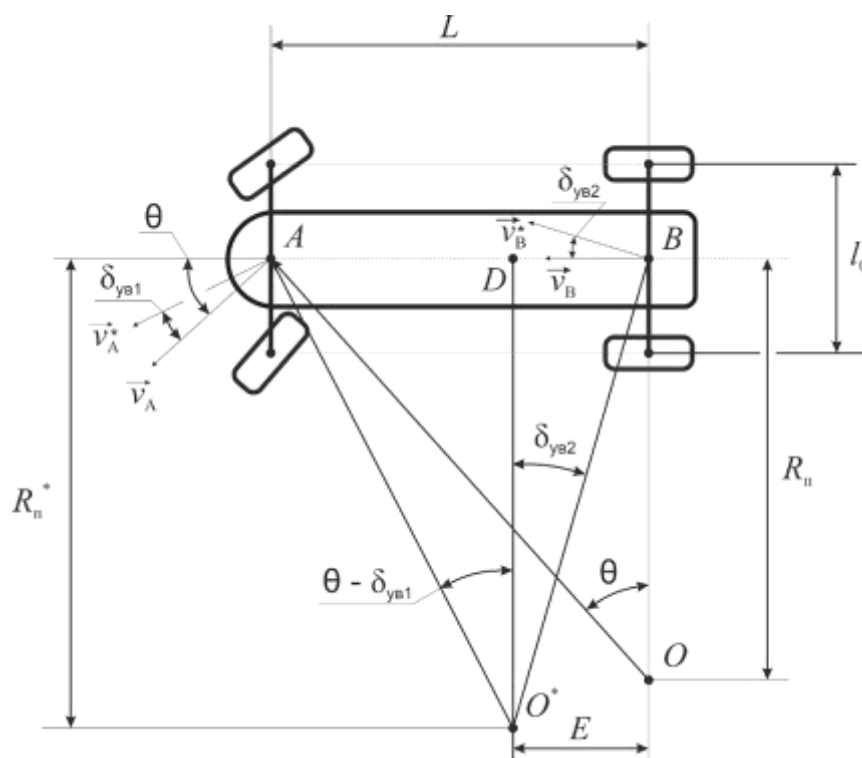


Рисунок 2.6 – Схема поворота ТС при качении колёс с уводом [46]

Векторы скоростей точек A и B без учёта бокового увода колёс обозначены \vec{v}_A и \vec{v}_B , а при уводе \vec{v}_A^* и \vec{v}_B^* . Из рисунка 2.6 непосредственно находим

$$AD = R_{\Pi}^* \operatorname{tg}(\theta - \delta_{\text{ув}1}); DB = R_{\Pi}^* \operatorname{tg}(\delta_{\text{ув}2}).$$

Так как

$$L = AD + DB, \text{ то}$$

$$R_{\Pi}^* = \frac{L}{\operatorname{tg}(\theta - \delta_{\text{ув}1}) + \operatorname{tg}(\delta_{\text{ув}2})}. \quad (2.5)$$

Углы увода колес обычно не превышают 5—10° [46]. Углы поворота управляемых колес при высоких скоростях движения также невелики, поэтому выражение (2.6) можно упростить:

$$R_{\Pi}^* = \frac{L}{\theta + (\delta_{\text{ув}2} - \delta_{\text{ув}1})}. \quad (2.7)$$

В зависимости от соотношения углов увода колес переднего и заднего мостов радиус поворота R_{Π}^* может быть равным радиусу поворота R_{Π} , без увода (при $\delta_{\text{ув}1} = \delta_{\text{ув}2}$), меньше (при $\delta_{\text{ув}1} < \delta_{\text{ув}2}$) или больше его (при $\delta_{\text{ув}1} > \delta_{\text{ув}2}$).

Мгновенный центр поворота при уводе колес всегда смещается внутрь базы ТС. Смещение E центра поворота O^* определяется по формуле

$$E = R_{\Pi}^* \operatorname{tg}(\delta_{\text{ув}2}) \cong R_{\Pi}^* \delta_{\text{ув}2}. \quad (2.8)$$

2.3 Математическая модель кинематического управления четырёхколёсным ТС

Рассмотрим математическую модель кинематического управления движением четырёхколёсным ТС, предложенную д-р физ.-мат. наук Е.А. Митюшевым, д-р физ.-мат. наук С.А. Брестовой и канд. физ.-мат. наук Ф.Д. Мисюра описанную

в публикациях [94, 95] и основанную на классических работах Е.А. Чудакова, и Й. Раймпеля [89, 91].

Рассмотрим кинематическую схему движения четырёхколёсного ТС приведённую на рисунке 2.7. Будем считать, что при движении ТС выполняется кинематическое условие Аккермана (Жанто) (2.2). Колёса вращаются без проскальзывания. Определим уравнения кинематического управления движением для данного ТС.

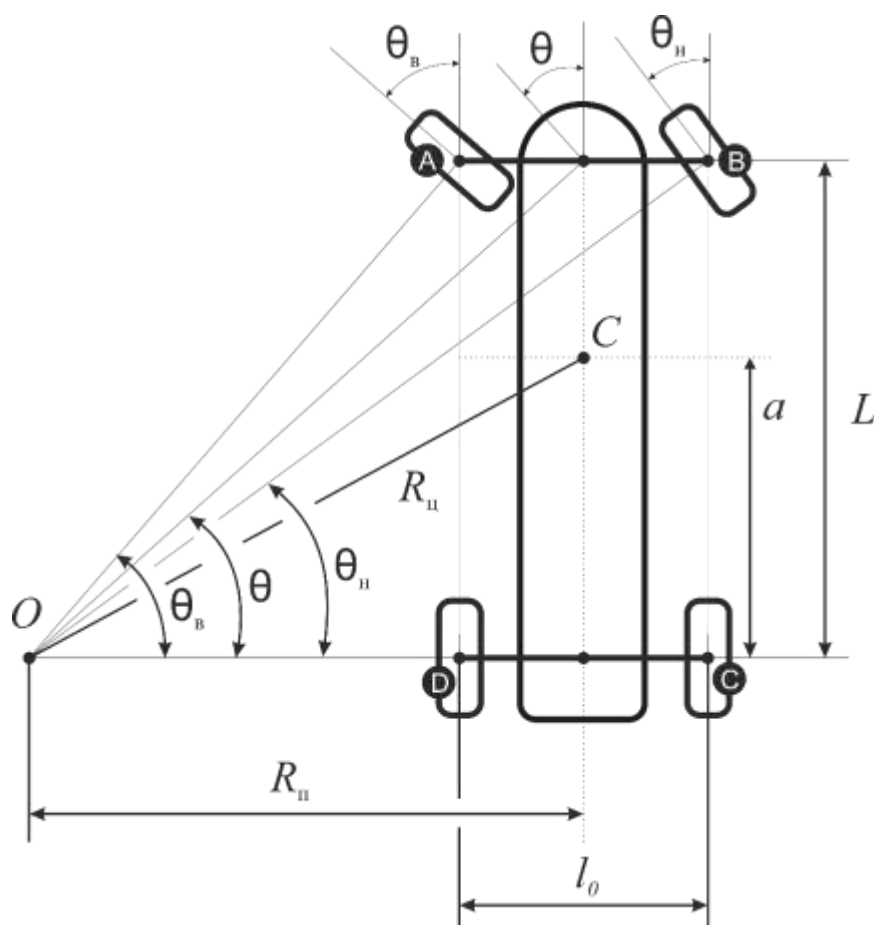


Рисунок 2.7 – Кинематическая схема движения четырехколесного транспортного средства. Источник иллюстрации [95].

где θ_B и θ_H — угол поворота управляемых колес внутреннего и наружного соответственно, которые определяются с учетом положения мгновенного центра скоростей O .

Рулевая трапеция Аккемана (Жанто) приведённая на рисунке 2.2 с достаточной степенью точности обеспечивает реализацию принципа Аккермана.

Таким образом окружность радиусом $R_{ц}$ будет соответствовать траектории движения центра масс ТС [94]:

$$R_{ц} = \sqrt{a^2 + L^2 \operatorname{ctg}^2 \theta}, \quad (2.8)$$

где θ определяется из формулы (2.4) и является углом эквивалентом углу поворота двухколёсной модели имеющей аналогичную базу L и осуществляющей движение по окружности радиуса $R_{п}$ (рисунок 2.8).

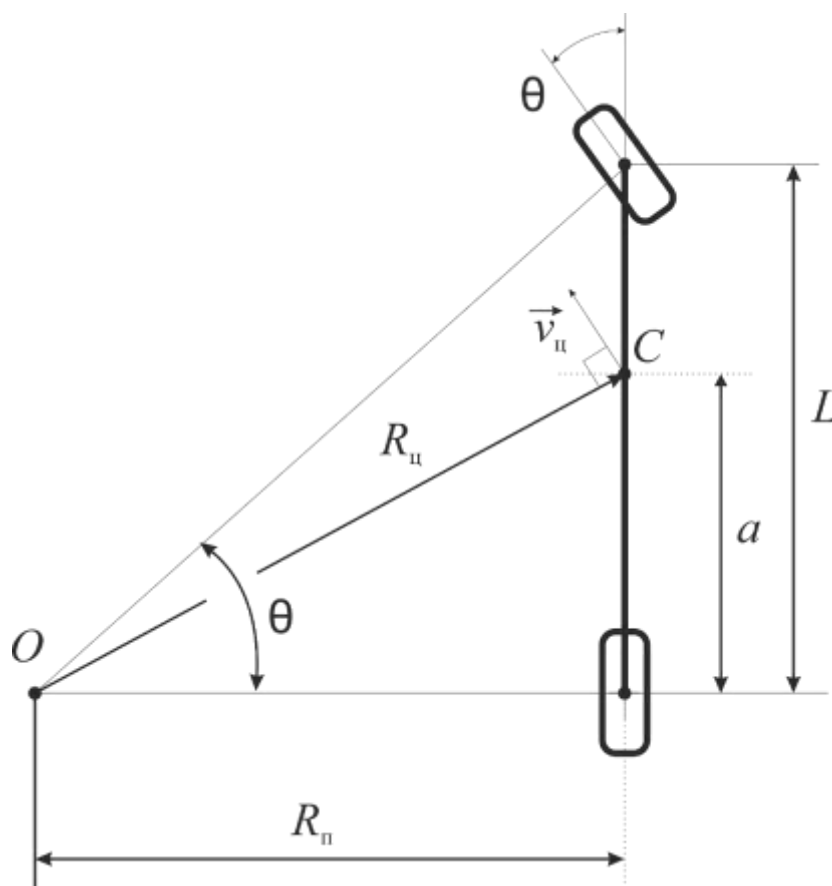


Рисунок 2.8 – Двухколёсная модель ТС. Источник иллюстрации [95].

Примем θ в качестве параметра управления и введём его в натуральное уравнение движения центра масс ТС [94]:

$$k = k(s)$$

где

$$k(s) = \frac{1}{R_{ц}}$$

Таким образом из (2.8) можно найти закон управления ТС по заданной траектории [94]:

$$\theta = \frac{k(s)}{|k(s)|} \operatorname{arcctg} \sqrt{\frac{\frac{1}{k^2(s)} - a^2}{L^2}} \quad (2.9)$$

При параметрическом задании с достаточно точным приближением можно определить, как равенство [94]:

$$k = \frac{\dot{x}\ddot{y} - \ddot{x}y}{\sqrt{(\dot{x}^2 + \dot{y}^2)^3}} \quad (2.10)$$

Тогда (2.9) с учётом (2.10) примет вид [94]:

$$\theta = \frac{\dot{x}\ddot{y} - \ddot{x}y}{|\dot{x}\ddot{y} - \ddot{x}y|} \operatorname{arcctg} \sqrt{\frac{\frac{(\dot{x}^2 + \dot{y}^2)^3}{(\dot{x}\ddot{y} - \ddot{x}y)^2} - a^2}{L^2}}. \quad (2.11)$$

Закон изменения углов поворота передних колес ТС с учетом (2.2) и (2.4) определяется как [94]:

$$\operatorname{ctg}\theta_{\text{в}} = \operatorname{ctg}\theta - \frac{l_0}{2L}, \quad \operatorname{ctg}\theta_{\text{н}} = \operatorname{ctg}\theta + \frac{l_0}{2L}.$$

Угловая скорость кузова ТС в его вращении вокруг вертикальной оси определяется равенством [94]:

$$\omega = \dot{\psi},$$

где ψ угол поворота касательной к траектории движения ТС.

С учетом определения кривизны траектории можно записать [94]:

$$\omega = \frac{d\psi}{dt} = \frac{d\psi}{ds} \frac{ds}{dt} = v_c k(s).$$

$$v_c = \sqrt{\dot{x}_c^2 + \dot{y}_c^2}. \quad (2.12)$$

Траектории точек касания колес определяются уравнениями [94]:

$$\begin{pmatrix} x_A \\ y_A \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} \cos\psi(t) & -\sin\psi(t) \\ \sin\psi(t) & \cos\psi(t) \end{pmatrix} \begin{pmatrix} x + (L - a) \\ y - \frac{l_0}{2} \end{pmatrix},$$

$$\begin{pmatrix} x_B \\ y_B \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} \cos\psi(t) & -\sin\psi(t) \\ \sin\psi(t) & \cos\psi(t) \end{pmatrix} \begin{pmatrix} x + (L - a) \\ y + \frac{l_0}{2} \end{pmatrix},$$

$$\begin{pmatrix} x_C \\ y_C \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} \cos\psi(t) & -\sin\psi(t) \\ \sin\psi(t) & \cos\psi(t) \end{pmatrix} \begin{pmatrix} x - a \\ y + \frac{l_0}{2} \end{pmatrix},$$

$$\begin{pmatrix} x_D \\ y_D \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} \cos\psi(t) & -\sin\psi(t) \\ \sin\psi(t) & \cos\psi(t) \end{pmatrix} \begin{pmatrix} x - a \\ y - \frac{l_0}{2} \end{pmatrix}.$$

Другой способ записи уравнений траекторий движения колес [94]:

$$\begin{pmatrix} x_A \\ y_A \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} + (L - a) \begin{pmatrix} \frac{\dot{x}}{\sqrt{\dot{x}^2 + \dot{y}^2}} \\ \frac{\dot{y}}{\sqrt{\dot{x}^2 + \dot{y}^2}} \end{pmatrix} - \frac{l_0}{2} \begin{pmatrix} \cos \frac{\pi}{2} & -\sin \frac{\pi}{2} \\ \sin \frac{\pi}{2} & \cos \frac{\pi}{2} \end{pmatrix} \begin{pmatrix} \frac{\dot{x}}{\sqrt{\dot{x}^2 + \dot{y}^2}} \\ \frac{\dot{y}}{\sqrt{\dot{x}^2 + \dot{y}^2}} \end{pmatrix},$$

$$\begin{pmatrix} x_B \\ y_B \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} + (L - a) \begin{pmatrix} \frac{\dot{x}}{\sqrt{\dot{x}^2 + \dot{y}^2}} \\ \frac{\dot{y}}{\sqrt{\dot{x}^2 + \dot{y}^2}} \end{pmatrix} + \frac{l_0}{2} \begin{pmatrix} \cos \frac{\pi}{2} & -\sin \frac{\pi}{2} \\ \sin \frac{\pi}{2} & \cos \frac{\pi}{2} \end{pmatrix} \begin{pmatrix} \frac{\dot{x}}{\sqrt{\dot{x}^2 + \dot{y}^2}} \\ \frac{\dot{y}}{\sqrt{\dot{x}^2 + \dot{y}^2}} \end{pmatrix}$$

$$\begin{pmatrix} x_c \\ y_c \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} - a \begin{pmatrix} \frac{\dot{x}}{\sqrt{\dot{x}^2 + \dot{y}^2}} \\ \dot{y} \\ \sqrt{\dot{x}^2 + \dot{y}^2} \end{pmatrix} + \frac{l_0}{2} \begin{pmatrix} \cos \frac{\pi}{2} & -\sin \frac{\pi}{2} \\ \sin \frac{\pi}{2} & \cos \frac{\pi}{2} \end{pmatrix} \begin{pmatrix} \frac{\dot{x}}{\sqrt{\dot{x}^2 + \dot{y}^2}} \\ \dot{y} \\ \sqrt{\dot{x}^2 + \dot{y}^2} \end{pmatrix}$$

$$\begin{pmatrix} x_c \\ y_c \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} - a \begin{pmatrix} \frac{\dot{x}}{\sqrt{\dot{x}^2 + \dot{y}^2}} \\ \dot{y} \\ \sqrt{\dot{x}^2 + \dot{y}^2} \end{pmatrix} - \frac{l_0}{2} \begin{pmatrix} \cos \frac{\pi}{2} & -\sin \frac{\pi}{2} \\ \sin \frac{\pi}{2} & \cos \frac{\pi}{2} \end{pmatrix} \begin{pmatrix} \frac{\dot{x}}{\sqrt{\dot{x}^2 + \dot{y}^2}} \\ \dot{y} \\ \sqrt{\dot{x}^2 + \dot{y}^2} \end{pmatrix}$$

Рассмотрим пример из [94]. Пусть траектория движения центра масс ТС, задана уравнениями:

$$x = 5t \text{ м}, \quad y = 5 - 5 \cos \frac{\pi}{2} t \text{ м.}$$

Габариты ТС:

$$l_0 = 1.5 \text{ м}, \quad L = 2 \text{ м}, \quad a = 1 \text{ м.}$$

В условиях рассматриваемого примера получим [94]:

$$\theta = \frac{\frac{25\pi^2}{4} \cos \frac{\pi}{2} t}{\left| \frac{25\pi^2}{4} \cos \frac{\pi}{2} t \right|} \operatorname{arcctg} \sqrt{\frac{\left(5^2 + \left(\frac{5\pi}{2} \sin \frac{\pi}{2} t \right)^2 \right)^3 - a^2}{\left(\frac{25\pi^2}{4} \cos \frac{\pi}{2} t \right)^2 L^2}},$$

График закона управления для рассматриваемого примера приведён на рисунке 2.9. Визуализация движения ТС выполнено в пакете MathCAD. На рисунке 2.10 показаны кадры анимации движения колес ТС для моментов времени: $t = 1 \text{ с}$, $t = 2 \text{ с}$, $t = 3 \text{ с}$, $t = 4 \text{ с}$.

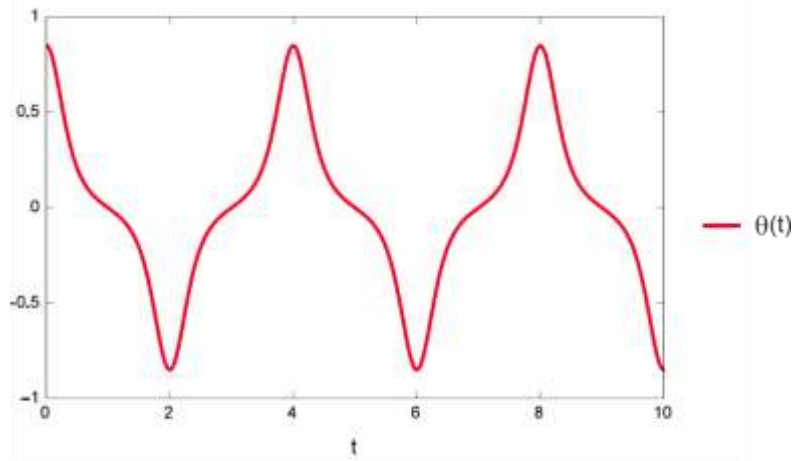


Рисунок 2.9 – График закона управления. Источник иллюстрации [95].

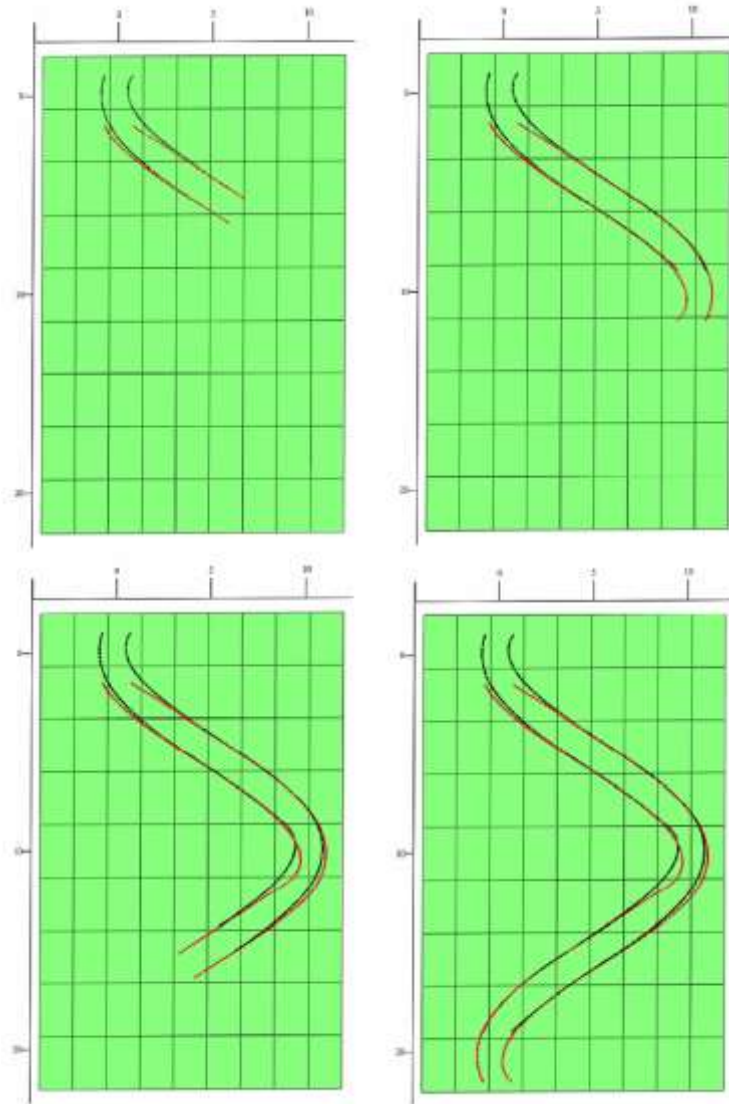


Рисунок 2.8. – Кадры анимации следов оставляемых ТС. Источник иллюстрации [94].

2.4 Выводы по главе 2

- В условиях решаемой задачи, влияние бокового увода, обусловленного динамическими параметрами АТМ пренебрежимо мало, так как скорости перемещения АТМ в рабочей зоне малы [46] (не превышают 1,5м/с).
- В силу того, что при малых скоростях движения принцип Аккермана выполняется с достаточной точностью [46], нет необходимости в применении более сложных моделей поворота (например, теории поворота Гауфа – Ширера [98], Фиала [99] или Хассельгрубера [100]).
- Рассмотренная математическая модель кинематического управления четырёхколёсным ТС, пригодна для использования в качестве основы для разработки системы управления траекторией АТМ функционирующих в условиях складского комплекса.

Глава 3. Разработка алгоритмического обеспечения системы формирования маршрута колёсного робота

Как было отмечено выше, метод опорных траекторий предполагает декомпозицию системы управления колёсным роботом, а именно разделение бортового ЦВС ричтрака на две взаимозависимых системы: во-первых, систему генерации опорной траектории, во-вторых, систему исполнения задания. Структурная схема реализации информационных потоков системы управления колёсным роботом, предложенная в рамках настоящего исследования, приведена на рисунке 3.1.

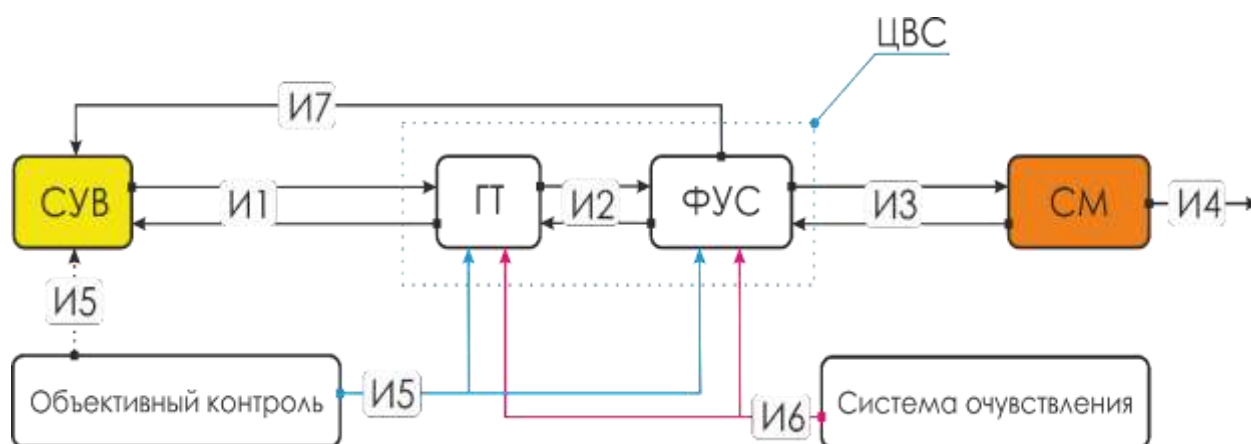


Рисунок 3.1 – Структурная схема информационного взаимодействия системы управления. СУВ – система управления верхнего уровня; ЦВС – центральная вычислительная система колёсного робота; ГТ – модуль генератора опорной траектории; ФУС – модуль формирования сигналов управления; СМ – сигнальный модуль; И1... И6 – направления передачи данных

Здесь СУВ – система управления верхнего уровня, осуществляющая определение цели перемещения робота. В нашем случае это интегрированная в инфраструктуру логистического центра вычислительная система, аналогичная системе GLV проекта PAN-Robots, рассмотренного ранее. СУВ, используя информационный канал И1, передаёт роботу данные о его пространственном расположении, а

также о заданном пространственном расположении относительно неподвижной системы координат, привязанной к территории складского комплекса. В качестве модуля-приёмника информации, переданной СУВ, выступает модуль генератора опорной траектории – ГТ. Так как СУВ является сторонней системой по отношению к КР, то такая передача может осуществляться по беспроводной сети, например, wi-fi.

ГТ, используя совокупность данных, полученных от СУВ, системы датчиков внешней информации (канал И6) и системы объективного контроля (канал И5), осуществляет планирование маршрута робота, которое заключается в определении некоторой кривой (являющейся аппроксимацией набора пролётных пространственных расположений), по которой необходимо пройти роботу для того, чтобы достичь указанной цели (планирование опорной траектории осуществляется относительно собственной системы координат, связанной с роботом). Отметим, что под системой объективного контроля подразумевается интегрированная в инфраструктуру логистического центра система слежения, аналогичная массиву потолочных лазерных сканеров, которые использованы при реализации проекта PAN-Robots.

Полученный маршрут по каналу И2 передаётся подсистеме формирования сигналов управления – ФУС, которая, используя встроенный алгоритм, преобразует маршрут в задание, представляющие собой упорядоченный набор углов поворота управляющих колёс, и затем через рассчитанные интервалы времени передаёт данные об углах поворота сигнальному модулю – СМ (канал И3). СМ осуществляет преобразование полученных от ФУС данных в электрические сигналы управления и передаёт их на исполнительные устройства мобильной платформы (канал И4), что проиллюстрировано в публикации [99].

Как только ФУС получает информацию от СМ о том, что задание выполнено (канал И3), используя данные систем датчиков внешней информации и объектив-

ного контроля (каналы И5 и И6), проводится верификация пространственного расположения робота относительно заданного пространственного расположения, после чего ФУС передаёт информацию о выполнении задания СУВ (канал И7).

В процессе перемещения ФУС использует данные канала И5 для контроля отклонения робота от маршрута и в случае, если такое отклонение фиксируется, проводит коррекцию набора пролётных точек задания. Также во время движения ФУС непрерывно анализирует информацию, получаемую от системы датчиков внешней информации, которая выступает в качестве сигнала защитной остановки в случае обнаружения препятствия (см. публикацию [100]). Если препятствие обнаружено, осуществляется СУВ обращается к ГТ за альтернативным маршрутом, позволяющим продолжить движение к заданному пространственному расположению и исключая столкновение с препятствием.

В рамках настоящего исследования считается, что СУВ, система датчиков внешней информации, система объективного контроля, а также СМ существуют. Поэтому сосредоточим внимание на подсистемах исполнения задания и планирования маршрута. Отметим, что для решения данной задачи необходимо, во-первых, определиться с представлением мобильной платформы как объекта управления, во-вторых, определить класс кривых, которые будут использоваться в качестве опорных траекторий, и, в-третьих, разработать алгоритмическое обеспечение для ГТ и ФУС подсистем.

3.1 Представления мобильной платформы в работе

В зависимости от выбранного подхода к решению задачи создания системы управления движением колёсного робота в аналогичных работах объект управления чаще всего представляют либо в виде материальной точки (характерно для случаев, когда управление движением осуществляется с помощью нейронной сети, что

показано в работе [101]), либо используют математические модели, аналогичные рассмотренной выше. Однако в рамках настоящего исследования применяется иное представление мобильной платформы, позволяющее использовать преимущества, обусловленные конструктивными особенностями выбранного типа транспор­тёров (см. техническую документацию VDI 2198, транспортёр ET1 [102]) снизить требования к вычислительным мощностям бортовой ЦВС, не потеряв при этом практической ценности полученных результатов.

В данном случае в качестве представления мобильной платформы используется геометрическая модель, изображённая на рисунке 3.2. Основная идея, положенная в основу данной модели, заключается в наличии жёсткой механической связи между точками касания колёс транспортёра и поверхности перемещения (покрытие рабочей зоны). Данная связь обусловлена особенностью конструкции всех четырехколёсных транспортных средств, имеющих автомобильную кинематическую схему с одной парой управляющих колёс.

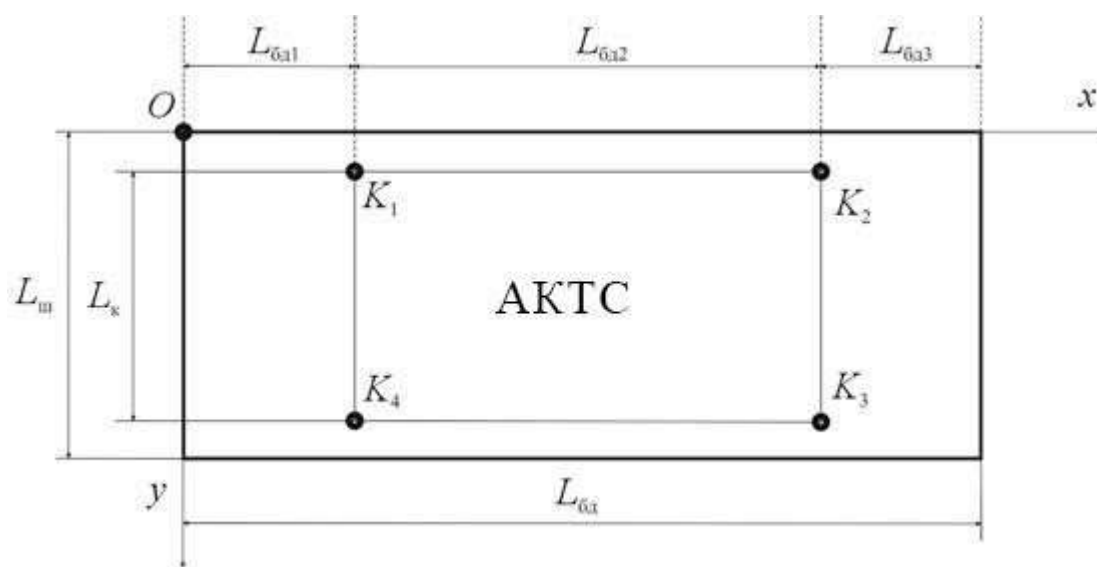


Рисунок 3.2 – Представление мобильной платформы в работе. Точки K_1 , K_2 , K_3 , K_4 – точки касания колёс мобильной платформы поверхности перемещения; $L_{бд}$, $L_{бд1}$, $L_{бд2}$, $L_{бд3}$, $L_{ш}$, $L_{к}$ – базовые геометрические размеры, определяемые конструкцией

КР; Источник иллюстрации [103]

Здесь K_1, K_2, K_3, K_4 – точки касания колёс мобильной платформы поверхности перемещения. При этом будем считать K_1, K_4 свободно катящимися симметричными колёсными модулями, K_2, K_3 , пара управляющих колёс – симметричные поворотные колёсные модули (типы колёсных модулей описаны в работе [104]). Большим прямоугольником обозначены габаритные размеры робототехнического комплекса, включающие все выступающие элементы конструкции, а также транспортируемый груз.

В дополнение к вышесказанному введём следующие модельные предположения для узлов мобильной платформы и поверхности перемещения:

- колёса являются абсолютно жёсткими,
- пятно касания поверхности перемещения колёсами мобильной платформы является точечным,
- поверхность перемещения представляет собой ровную, шероховатую плоскость, обеспечивающую качение колёс без проскальзывания,
- эксцентриситет управляющих колёс отсутствует,
- поворот управляющих колёс на заданный угол осуществляется мгновенно.

Также наложим ограничения на скорость движения, а именно, предположим, что скорость движения робота по территории логистического центра не превышает 10 км/ч.

Отметим, что во всех расчётах, выполненных в рамках настоящего исследования, в качестве массогабаритных характеристик использованы параметры транспортёра ET1 производства компании БАЛКАНКАР РЕКОРД АО (см. техническую документацию VDI 2198, транспортёр ET1 [102]). Выбор данного транспортёра обусловлен тем, что, во-первых, он имеет интересующую нас конструкцию, во-вторых, является широко распространённым на отечественных машиностроительных предприятиях.

3.2 Опорная траектория. Кривые Безье

Следующим этапом разработки алгоритмического обеспечения для модулей генератора траектории и формирования сигналов управления является определение типа кривых, которые будут использоваться в качестве опорных траекторий. Причём выбранный тип кривых должен обеспечивать возможность предсказуемого манипулирования результирующей кривой и легко поддаваться машинной обработке.

В рамках настоящей работы в качестве потенциальных опорных траекторий рассматривались В-сплайны, описанные в работах [105, 106], кубические сплайны и кривые Безье (см. [107]). В результате проведённого анализа выбор был сделан в пользу кривых Безье, на что имеется несколько причин. Во-первых, конечная форма кривой Безье зависит только от взаимного расположения вершин опорного многоугольника (следует из их математической основы – базиса Бернштейна, что показано в работах [108, 109]), что позволяет легко управлять формой кривой, а также обеспечивает предсказуемость изменения её формы. Таким образом, задача планирования опорной кривой сводится к определению координат вершин опорного многоугольника, при которых полученная кривая будет удовлетворять требуемым условиям, наложенным на опорную траекторию, что показано в публикации [103]. Как следствие, передача информации об опорной траектории между ГТ и ФУС значительно упрощается, так как сводится к передаче набора из n пар чисел, определяющих опорную кривую. Во-вторых, большинство развитых языков программирования имеют готовые хорошо оптимизированные решения, реализующие работу с данным типом кривых, т.к. они имеют широкое распространение в компьютерной графике, что снимает необходимость в разработке дополнительного программного обеспечения (библиотек) для выполнения операций над опорными траекториями.

«Математическое параметрическое представление кривой Безье имеет вид [108]:

$$P(t) = \sum_{i=0}^n B_i J_{n,i}(t), \quad 0 \leq t \leq 1, \quad (3.1)$$

где B_i — i -ый вертекс полигона, $J_{n,i}$ базис Безье или Бернштейна, или функция аппроксимации [108]:

$$J_{n,i}(t) = \binom{n}{i} t^i (1-t)^{n-i} \quad (3.2)$$

с

$$\binom{n}{i} = \frac{n!}{i!(n-i)!} \quad (3.3)$$

$J_{n,i}(t)$ — i -я функция базиса Бернштейна порядка n . Здесь n – порядок определяющей функции базиса Бернштейна – и, следовательно, сегмента полиномиальной кривой, на единицу меньше количества точек определяющего многоугольника.» [109] Роджерс Д. и др., Математические основы машинной графики, с. 297-298, (В оригинале, формулы 3.1, 3.2, 3.3 имеют нумерацию 5-62, 5-63 и 5-64 соответственно).

В качестве опорных кривых будем использовать кривые Безье третьего порядка, что обусловлено геометрией траекторий движения четырехколёсных мобильных роботов, имеющих автомобильную кинематическую схему с одной парой управляющих колёс. Тогда базисные функции будут иметь вид (3.4):

$$\begin{aligned} J_{3,0}(t) &= (1-t)^3 \\ J_{3,1}(t) &= 3t(1-t)^2 \\ J_{3,2}(t) &= 3t^2(1-t) \\ J_{3,3}(t) &= t^3 \end{aligned} \quad (3.4)$$

Следовательно, уравнение для определения координат точек на кривой (3.1) для рассматриваемого случая с учётом (3.4) будет иметь вид (3.5):

$$P(t) = B_0(1-t)^3 + B_1 3t(1-t)^2 + B_2 3t^2(1-t) + B_3 t^3 \quad (3.5)$$

Специфика алгоритма исполнения задания, предложенного в данной работе, помимо расчёта координат точек на опорной кривой требует определения касательных к опорной кривой в пролётных точках задания в виде $y = f(t)$. Найдём уравнения касательных для кривой Безье третьего порядка в заданной точке, для чего необходимо взять первую производную от исходной кривой, уравнение которой в общем виде имеет вид (3.6) [108]:

$$\dot{P}(t) = \sum_{i=0}^n B_i \dot{J}_{n,i}(t) \quad (3.6)$$

Продифференцируем базисные функции 3.4 по параметру t и получим:

$$\begin{aligned} \dot{J}_{3,0}(t) &= -3(1-t)^2 \\ \dot{J}_{3,1}(t) &= 3(1-t)^2 - 6t(1-t) \\ \dot{J}_{3,2}(t) &= 6t(1-t) - 3t^2 \\ \dot{J}_{3,3}(t) &= 3t^2. \end{aligned} \quad (3.7)$$

Подставим (3.7) \rightarrow (3.6), получим уравнение для определения точек на кривой, являющейся производной от исходной кривой:

$$\begin{aligned} \dot{P}(t) &= -3(1-t)^2 B_0 + B_1(3(1-t)^2 - 6t(1-t)) \\ &+ B_2(6t(1-t) - 3t^2) + B_3 3t^2. \end{aligned} \quad (3.8)$$

что показано в публикации [75].

На рисунке 3.3 проиллюстрирована геометрическая интерпретация производной кривой Безье. Отдельно отметим, что, как это показано в публикации [107], производная кривой Безье обладает следующим свойством: угол наклона прямой, проведённой через точки $[0,0]$ и точку $P(t_i)$ на кривой, соответствует углу наклона касательной к исходной кривой в точке $P(t_i)$.

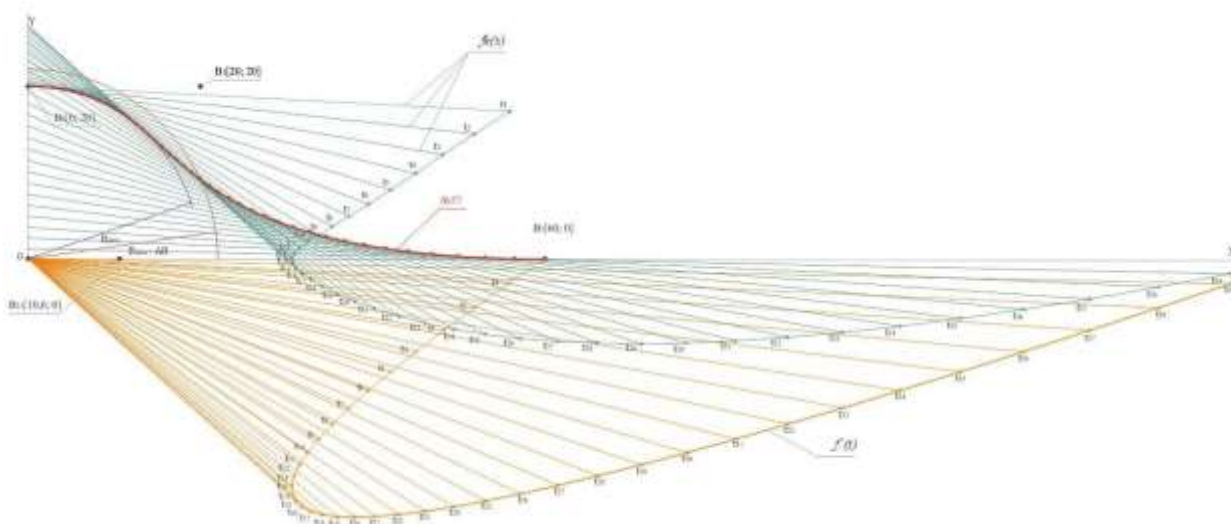


Рисунок 3.3 – Кривая Безье и её первая производная, иллюстрация создана по материалам публикации [107]

Запишем уравнение прямой, проходящей через 2 точки, в декартовой системе координат:

$$\frac{y - y_1}{y_2 - y_1} = \frac{x - x_1}{x_2 - x_1}. \quad (3.9)$$

Далее, используя (3.8) для любых значений t , можно определить координаты x_2 и y_2 , тогда как координаты x_1 и y_1 зафиксированы в точке $[0, 0]$. Подставляя найденные значения в (3.9), получим уравнение прямой, проходящей через начало координат и рассчитанную точку на кривой, являющейся производной от исходной кривой Безье $P(t_i)$:

$$y = xk = x \frac{\dot{P}_y(t_i)}{\dot{P}_x(t_i)} \quad (3.10)$$

Классическое уравнение прямой имеет вид:

$$y = kx + b. \quad (3.11)$$

Тогда уравнение прямой, проходящей через i -ую известную точку на исходной кривой, можно записать в виде:

$$P_y(t_i) = kP_x(t_i) + b. \quad (3.12)$$

Подставляя значение k из уравнения (3.10) в уравнение (3.12), получим формулу для определения b .

$$b = P_y(t_i) - \frac{\dot{P}_y(t_i)}{\dot{P}_x(t_i)} P_x(t_i). \quad (3.13)$$

Теперь, когда все коэффициенты определены, можно записать уравнение касательной, проходящей через i -ую точку исходной кривой Безье, в виде $y=f(t)$ для декартовой системы координат:

$$y = \frac{\dot{P}_y(t_i)}{\dot{P}_x(t_i)} (x - P_x(t_i)) + P_y(t_i). \quad (3.14)$$

В случае генерации опорной траектории, определяющей маршрут колёсного робота, приходится иметь дело не с одной кривой, а осуществлять выбор наиболее подходящей кривой из некоторого семейства кривых, каждая из которых определена четырьмя точками, заданными на некотором множестве точек. Определение наилучшей альтернативы будем осуществлять из соображений соответствия кривой заданным ограничениям, а также общей длины кривой, от которой зависит время выполнения задания. Откуда следует, что нам потребуется легко реализуемый с программной точки зрения способ оценки длины кривой, например, описанной в публикации [110].

Предположим, что для исследования кривой выбран такой шаг Δt параметра t , что участок кривой на интервале $t_i, t_i + \Delta t$ может быть заменён отрезком прямой, проходящей через точки $P(t_i)$ и $P(t_i + \Delta t)$, тогда длина i -ого отрезка может быть определена по формуле (3.15):

$$L_{i,i-1} = \sqrt{(P_x(t_i + \Delta t) - P_x(t_i))^2 + (P_y(t_i + \Delta t) - P_y(t_i))^2}, \quad (3.15)$$

Длина всей кривой, будет определяться как:

$$L_p = \sum_{i=0}^n \sqrt{(P_x(t_i + \Delta t) - P_x(t_i))^2 + (P_y(t_i + \Delta t) - P_y(t_i))^2}. \quad (3.16)$$

Теперь, имея в своём распоряжении геометрическую модель мобильной платформы и аналитическое представление опорной траектории, мы можем перейти к разработке алгоритмического обеспечения модуля ФУС.

3.3 Алгоритм исполнения задания

В большинстве случаев, как это показано в работах [111, 112], задача исполнения задания колёсным роботом рассматривается как задача оптимального управления (постановка задачи оптимального управления показана в работах [113, 114]), решение которой осуществляется при помощи сложных динамических моделей, определяющих связи между центром масс, программной траекторией движения и углом поворота управляющих колёс. Однако в рамках настоящего исследования применён другой подход, основными отличиями которого являются, во-первых, использование простого геометрического представления мобильной платформы как объекта управления, во-вторых, базисная точка системы координат мобильной платформы. В случае классического подхода базисная точка мобильной платформы, как правило, совпадает с положением центра масс робота, в данном же случае она смещена и привязана к точке касания неповоротного колесного модуля поверхности перемещения. Это позволяет в полной мере использовать преимущества, обусловленные особенностями конструкции четырехколёсных мобильных роботов с автомобильной кинематической схемой, описанными в работах [115, 116].

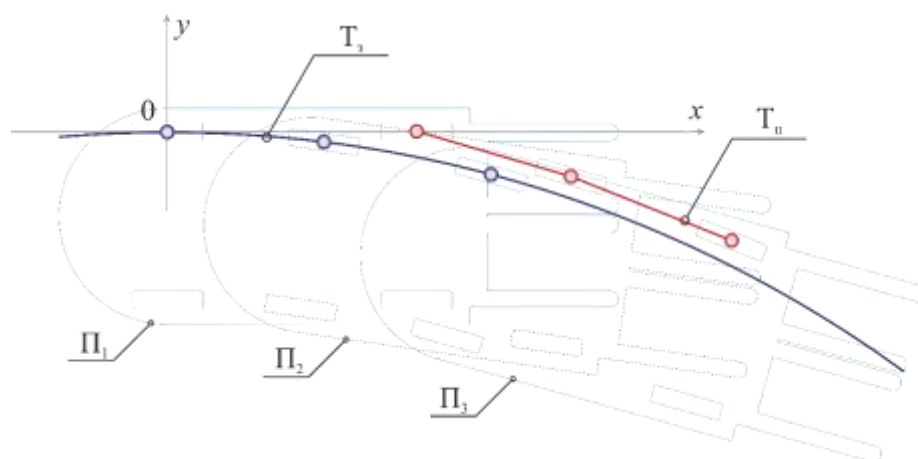


Рисунок 3.4 – «Следы» колёсных модулей мобильной платформы, расположенных на одной продольной оси. T_3 – след неповоротного колёсного модуля; T_n – след поворотного колёсного модуля; Π_1, Π_2, Π_3 – пространственные расположения робота на маршруте

Идея, положенная в основу предложенного алгоритма исполнения задания, заключается в том, что в силу наличия жёстких механических связей между точками касания колёс мобильной платформы поверхности перемещения, траектории, по которым будут двигаться колёса, находящиеся на одной продольной оси (по одному борту), связаны однозначным образом. Поэтому, зная траекторию, по которой «прошёл» неповоротный колёсный модуль (в нашем случае заднее колесо робота), всегда можно с требуемой точностью восстановить траекторию, по которой двигался поворотный (управляющий) колёсный модуль (рисунок 3.4). Руководствуясь этим принципом, опишем алгоритм, позволяющий определить управляющие воздействия для каждой пролётной точки задания на опорной траектории, где под опорной траекторией будем понимать траекторию движения неповоротного колёсного модуля.

Введём подвижную систему координат $x_10_1y_1$, базисную точку которой привяжем к точке касания поверхности перемещения (далее по тексту будем использовать термин – точка касания) неповоротным колёсным модулем мобильной платформы. Для того, чтобы избежать неопределённости, будем считать, что

точка отсчёта введённой системы координат на момент начала движения совпадает с положением точки касания заднего колеса мобильной платформы, расположенного по левому борту, которое в дальнейшем будем называть – задним колесом. Предположим, что ось x_1 введённой системы координат проходит через точки касания колёсами мобильной платформы, находящимися на одной продольной оси, т.е. задним колесом и передним поворотным колёсным модулем (в дальнейшем будем называть его – передним колесом), как показано на рисунке 3.5. Направление оси x_1 совпадает с направлением движения робота.

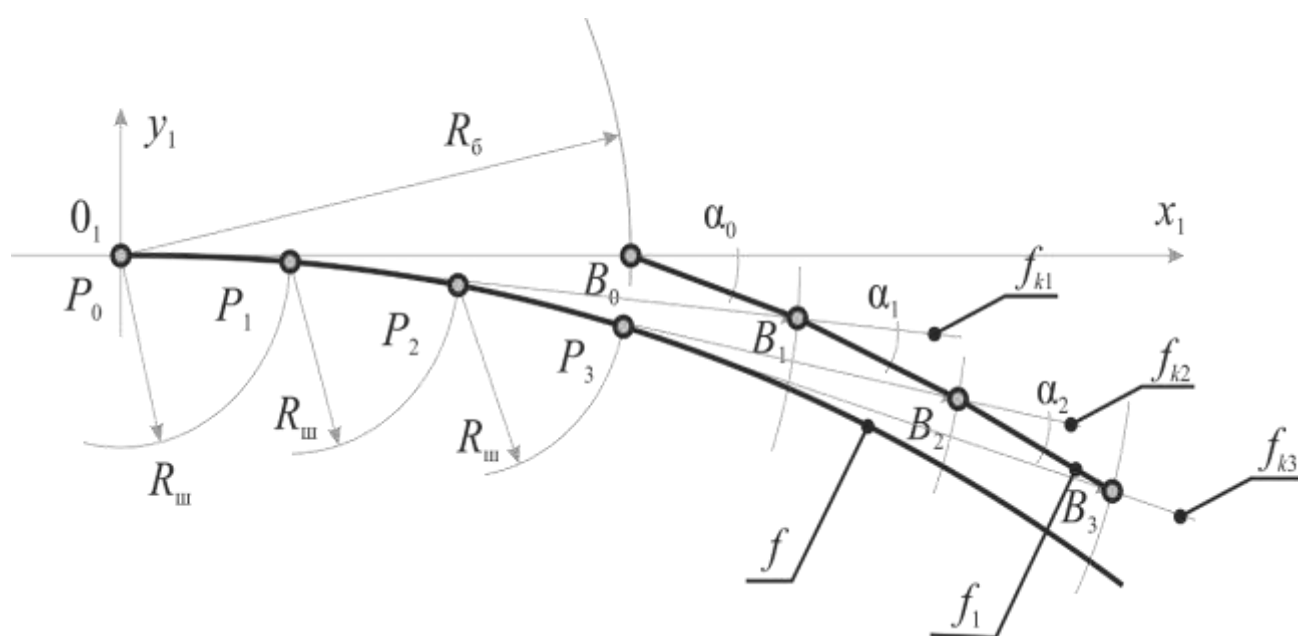


Рисунок 3.5 – Геометрическая иллюстрация алгоритма выполнения задания. Мобильная платформа движется вперёд; f – опорная траектория; f_1 – кусочно-линейная аппроксимация траектории переднего колеса; f_{k1}, f_{k2}, f_{k3} – касательные к опорной траектории; P_0, P_1, P_2, P_3 – положение заднего колеса на опорной траектории при перемещении мобильной платформы; B_0, B_1, B_2, B_3 – точки касания переднего колеса; $R_{ш}$ – шаг по опорной траектории; $R_б$ – базовый радиус; $\alpha_1, \alpha_2, \alpha_3$ – углы отклонения переднего колеса от продольной оси мобильной платформы

Пусть определена функция f , такая, что для каждой точки на отрезке $[x_0; x_{P_n}]$ можно однозначно определить касательную f_{ki} , где под отрезком $[x_0; x_{P_n}]$ будем понимать интервал f , совпадающий с траекторией заднего колеса мобильной платформы, т.е. являющийся опорной траекторией. Дополнительно потребуем, чтобы на интервале $[x_0; x_{P_n}]$ кривизна f не превышала минимального радиуса разворота робота $R_{min} = f(v)$, (понятие минимального радиуса разворота приведено в работе [46]), что обусловлено принципиальной возможностью «выполнить» перемещение по данной траектории (см. работу [117]).

В произвольной точке P_i , такой, что $P_{xi} \in [x_0; x_{P_n}]$, проведём окружность базового радиуса - R_6 с центром P_i , где R_6 есть расстояние между точками касания заднего и переднего колёс мобильной платформы. Затем проведём касательную f_{ki} в точке P_i . Следующим шагом найдём точки пересечения проведённых окружности и касательной. Принимая во внимание тот факт, что с учётом модельных допущений точка касания переднего колеса всегда находится на касательной к траектории заднего колеса, одна из найденных точек, а именно расположенная по ходу движения робота, будет определять положение переднего колеса, соответствующее текущему положению заднего колеса. Следовательно, если обойти интервал $[x_0; x_{P_n}]$ с некоторым шагом, то можно получить набор точек B , отражающих положение переднего колеса для каждой точки P , определяющей положение заднего колеса. В предельном случае – стягивая размер шага к нулю, можно получить точную аппроксимацию траектории переднего колеса. Зная положения переднего колеса и полагая, что в момент начала движения угол отклонения переднего колеса от продольной оси мобильной платформы известен, можно получить набор управляющих воздействий в виде углов отклонения переднего колеса от продольной оси мобильной платформы для каждого шага по кривой, т.е. набор пространственных расположений, которые необходимо выполнить, чтобы заднее колесо мобильной платформы переместилось по опорной траектории [110].

Пусть функция f – опорная траектория, задана кривой Безье третьего порядка $P(t)$ и имеет вид (3.17). Значение параметра t является безразмерной величиной, имеет физический смысл временного интервала, за который робот исполняет задание, и изменяется в пределах $[0,1]$.

$$P(t) = (1 - t)^3 P_0 + 3t(1 - t)^2 P_1 + 3t^2(1 - t) P_2 + t^3 P_3, t \in [0,1] \quad (3.17)$$

Здесь $P(t)$ – функция компонент опорных вершин в точке t_i . Используя представление опорной траектории (3.17), мы можем определить координаты x, y точки P_i для любого t на всём интервале $t \in [0,1]$ (рисунок 3.5). Производная кривой Безье третьего порядка, нахождение которой необходимо для определения касательных кривой в точках P_i , определяется уравнением (3.8). Будем двигаться по опорной траектории с шагом Δt , таким, что:

$$\Delta S(\Delta t) \ll R_6, R_6 = const \quad (3.18)$$

где ΔS – длина перемещения по опорной траектории за время Δt . Используя систему уравнений (3.19), для каждого шага определим положения точки касания переднего колеса относительно введённой системы координат как пересечение касательной к опорной траектории, заданной уравнением (3.17) и окружности R_6 с центром в точке P_i :

$$\begin{cases} (x - P(t_i)_x)^2 + (y - P(t_i)_y)^2 = R_6^2 \\ y(x) = P(t_i)_y + \frac{\dot{P}(t_i)_y}{\dot{P}(t_i)_x} (x - P(t_i)_x) \end{cases} \quad (3.19)$$

Угол отклонения переднего колеса от продольной оси мобильной платформы для каждого i -ого шага определим как угол между прямой, проходящей через точки B_i и B_{i+1} , и касательной к кривой описанной задним колесом f_{ki} в точке P_i и назовём α_i . Тогда угол «доворота» переднего колеса – β_i , т.е. угол, на который нужно повернуть переднее колесо, чтобы перевести его из положения, соответствующего углу α_i , в положение α_{i+1} , и будет искомым управляющим воздействием на i -ом шаге.

Чтобы установить аналитическую зависимость для определения управляющих воздействий, запишем уравнение (3.11) прямой, проходящей через точки $B(t_i)B(t_{i+1})$, с учётом (3.13) и (3.14), в виде (3.20).

$$y = x \left(\frac{B_y(t_{i+1}) - B_y(t_i)}{B_x(t_{i+1}) - B_x(t_i)} \right) + \left(B_y(t_i) - \left(\frac{B_x(t_i) (B_y(t_{i+1}) - B_y(t_i))}{B_x(t_{i+1}) - B_x(t_i)} \right) \right) \quad (3.20)$$

Тангенс угла между двумя пересекающимися прямыми определяется как:

$$tg(\alpha) = \frac{k_2 - k_1}{1 + k_1 k_2} \quad (3.21)$$

Следовательно, угол α_i на каждом шаге, используя (3.14) и (3.20), можно определить из уравнения (3.22), а управляющее воздействие на каждом шаге будет выражаться зависимостью (3.23).

$$\alpha_i = arctg \left(\frac{\left(\frac{B_y(t_{i+1}) - B_y(t_i)}{B_x(t_{i+1}) - B_x(t_i)} \right) - \frac{\dot{P}_y(t_i)}{\dot{P}_x(t_i)}}{1 + \left(\frac{B_y(t_{i+1}) - B_y(t_i)}{B_x(t_{i+1}) - B_x(t_i)} \right) \frac{\dot{P}_y(t_i)}{\dot{P}_x(t_i)}} \right) \quad (3.22)$$

$$\beta_i = \alpha_{i+1} - \alpha_i \quad (3.23)$$

Блок схема программы, реализующей предложенный алгоритм исполнения задания, приведена на рисунке 3.6.

Заметим, что точность кусочно-линейной аппроксимации траектории переднего колеса и, как следствие, качество отработки траектории напрямую зависит от N -количества шагов по кривой опорной траектории, что проиллюстрировано на рисунке 3.7. Также опытным путём установлено, что при $N < 20$ и скорости движения, не превышающей 5км/ч, во время движения мобильной платформы наблюдается эффект «рыскания», однако при $N = 40$ данный эффект исчезает, робот движется плавно.

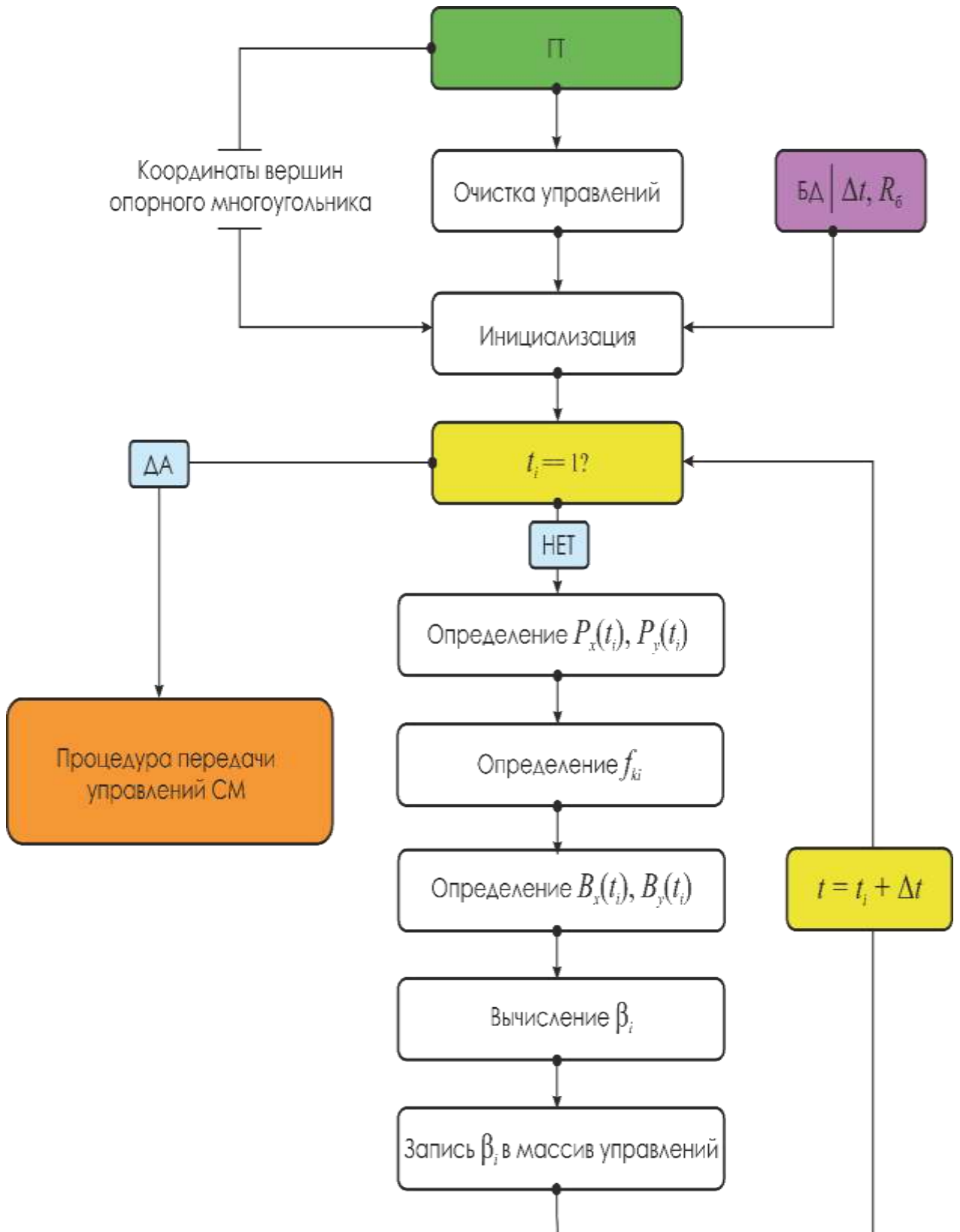


Рисунок 3.6 – Блок-схема алгоритма программы исполнения задания колёсным роботом

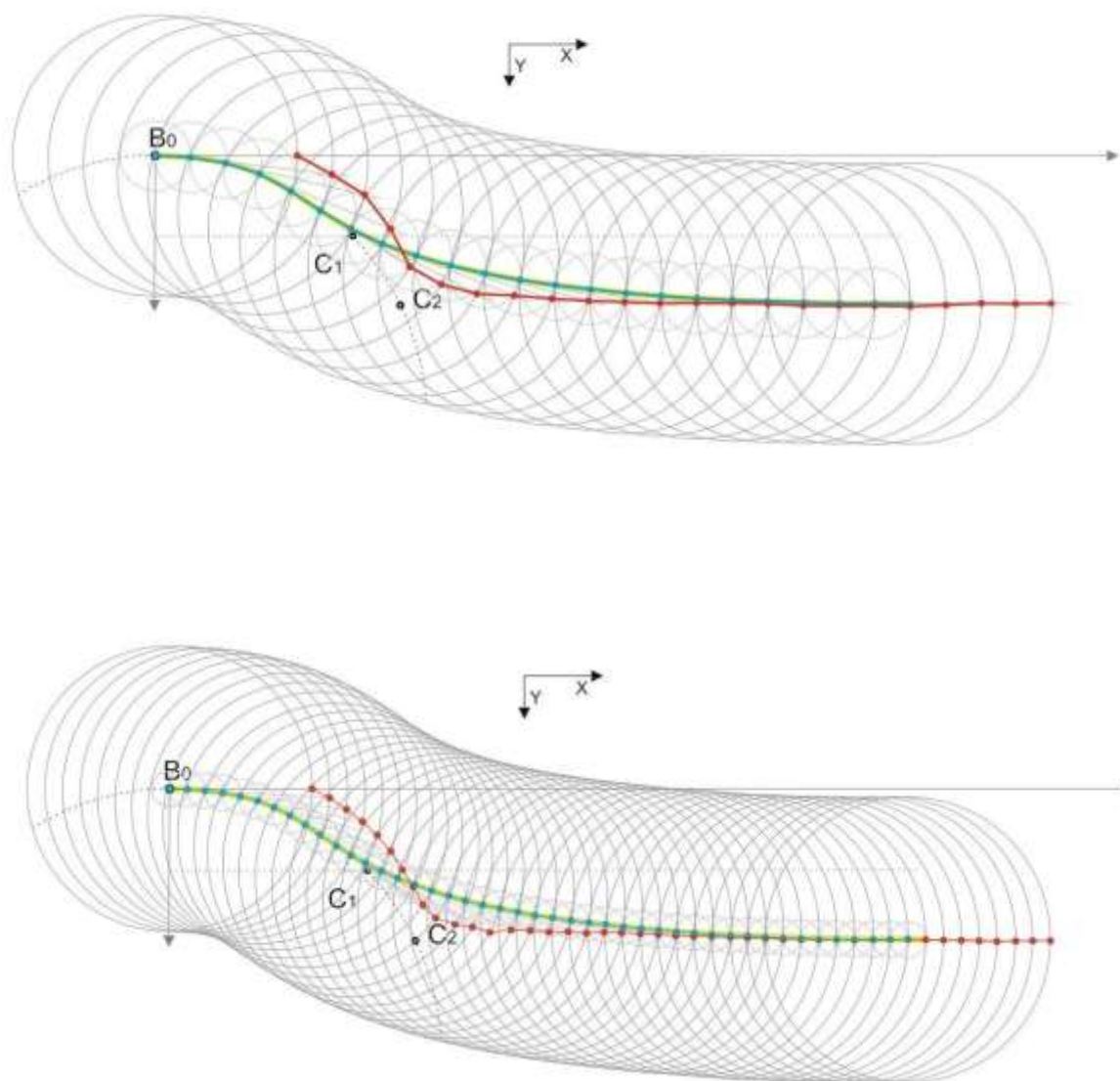


Рисунок 3.7 – Зависимость точности кусочно-линейной аппроксимации от количества шагов алгоритма исполнения задания $N=22$ (сверху), $N=44$ (снизу)

Разработанный алгоритм позволяет колёсному роботу осуществлять обработку криволинейных траекторий, заданных в виде кривых Безье третьего порядка, и является основой алгоритмического обеспечения модуля ФУС системы управления. Следует заметить, что данный алгоритм может быть использован как в случае движения мобильной платформы в прямом, так и в обратном направлении, что будет показано экспериментально [118].

Разработанный алгоритм не учитывает динамических характеристик мобильной платформы, влияющих на положение в пространстве во время движения по криволинейной траектории (поперечный увод), однако, принимая во внимание ограничение скоростей движения транспортёров по территории логистического центра, оговоренные выше, ошибка, связанная с динамикой мобильной платформы, оказывается пренебрежимо мала, а при необходимости её учета компенсация отклонения от желаемой траектории движения может осуществляться на основе информации, полученной от интегрированной в инфраструктуру логистического центра системы объективного контроля.

3.4 Алгоритмическое обеспечение модуля генератора опорной траектории

Сложность задачи определения опорной траектории заключается в том, что в силу особенностей выбранного типа кривых существует множество вариантов маршрутов для одной и той же задачи перемещения, где под задачей перемещения понимается перемещение мобильной платформы из точки A в точку B рабочей зоны. То есть существует некоторое в общем случае бесконечное множество альтернативных кривых, удовлетворяющих условиям текущей задачи перемещения с учётом всех имеющихся ограничений, обусловленных наличием как статических, так и динамических препятствий. Таким образом, задача генерации опорной траектории преобразуется в задачу поиска наилучшей альтернативы на заданном множестве, из которого требуется выбрать опорную траекторию, причём только одну, что предполагает пошаговое сужение области поиска.

В связи с этим в рамках настоящего исследования было предложено произвести декомпозицию задачи поиска опорной траектории и решать её поэтапно. Вследствие этого алгоритм работы модуля ГТ (генератора опорной траектории)

получил трёхуровневую блочную организацию, схематическое изображение которой приведено на рисунке 3.8.

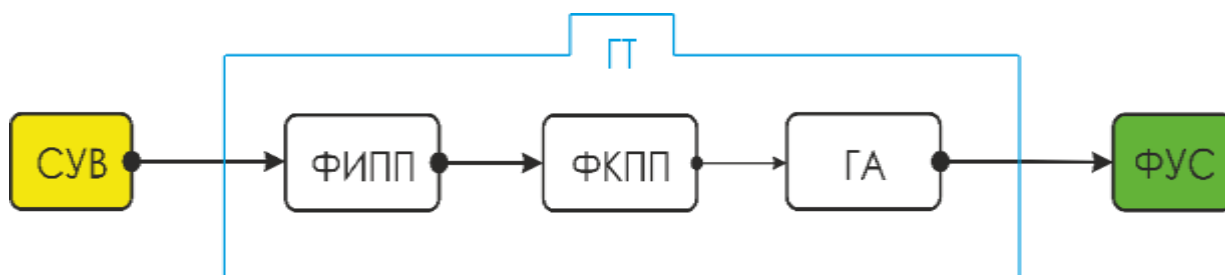


Рисунок 3.8 – Внутренняя организация модуля генератора опорных траекторий - ГТ; СУВ – система управления верхнего уровня; ФУС – модуль формирования сигналов управления; ФИПП – блок формирования исходного пространства поиска; ФКПП – блок формирования конечного пространства поиска; ГА – генетический алгоритм

Так блок ФИПП на основании задачи перемещения, поставленной системой управления верхнего уровня (СУВ), осуществляет формирование исходного пространства поиска в виде множества кривых, удовлетворяющих текущей задаче перемещения. Полученное исходное пространство поиска поступает на вход блока ФКПП, где в соответствии с установленным набором правил осуществляется выбраковка кривых, тем самым формируется конечное пространство поиска. Конечное пространство поиска передаётся на вход генетического алгоритма, с его помощью из конечного пространства поиска осуществляется выбор единственной кривой, которая в качестве опорной траектории передаётся модулю ФУС.

Следует заметить, что задача генерации опорных траекторий эквивалентна задаче поиска координат вершин многоугольника, определяющего форму кривой Безье третьего порядка относительно системы координат мобильной платформы, следовательно, все исследуемые пространства поиска будут представлять собой множества точек, определённых на плоскости.

3.5 Формирование исходного пространства поиска

Рассмотрим самый общий случай маневрирования колёсного робота, когда на пути его существует некоторое динамическое препятствие, тогда желаемая траектория движения робота, будет иметь вид, изображённый на рисунке 3.9.

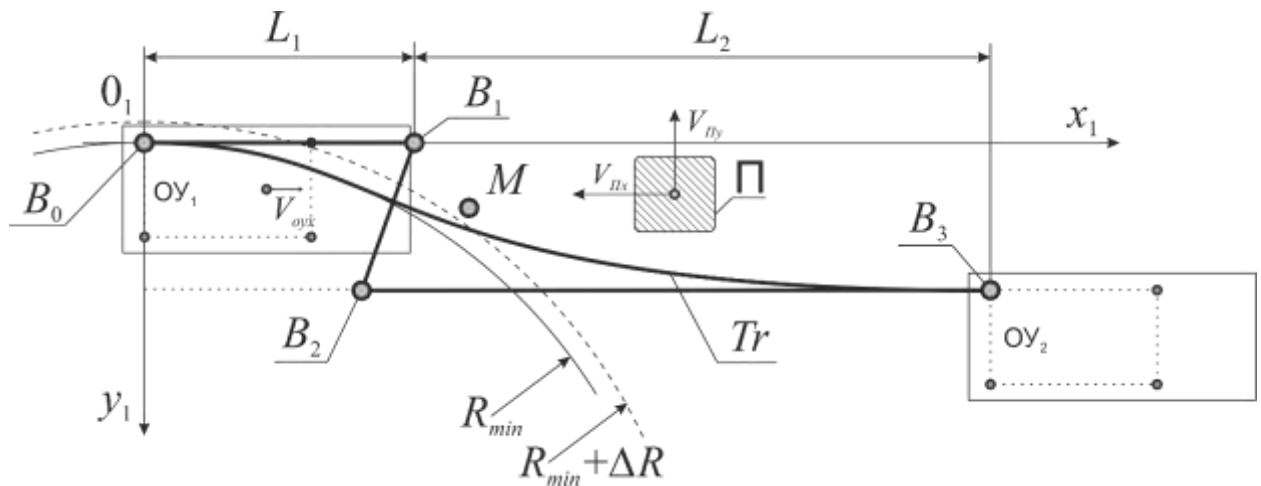


Рисунок 3.9. – Маневрирование. $x_1O_1y_1$ – система координат мобильной платформы; OY_1 – начальное пространственное расположение; OY_2 – заданное пространственное расположение; Π – препятствие; M – точка гипотетического столкновения; B_0, B_1, B_2, B_3 – вершины опорного многоугольника; Tr – кривая опорной траектории. Иллюстрация основана на публикации [103]

Проведя изучение значительного числа схем маневрирования, в рамках настоящей работы был сделан вывод, что решающее значение для успешного ухода от столкновения с динамическим препятствием имеет не общее время выполнения манёвра, а скорость прохождения роботом участка траектории от точки начального пространственного расположения (точка B_0) до точки гипотетического столкновения, на рисунке обозначенной как M . Таким образом, траектория движения робота при объезде препятствия может быть условно разделена на два участка, где первый участок назовём фазой активного уклонения, а второй

участок, на котором происходит выход из манёвра, – релаксацией (фазой релаксации).

Сформулируем требования, которые будем предъявлять к кривой, выступающей в качестве опорной для задания маршрута, исключающего столкновение с движущимся препятствием.

Потребуем, чтобы для всех точек, принадлежащих опорной кривой, значения y -координаты, отличались от y -координат траектории, соответствующей минимальному радиусу разворота мобильной платформы, не более чем на ΔR , то есть:

$$R_{min} \leq y(x) \leq R_{min} + \Delta R, \text{ при } x \in [0, D_x], \Delta R \ll L_{ш}. \quad (3.24)$$

К участку траектории, соответствующему фазе релаксации, никаких особых требований предъявлять не будем. [103]

Обозначим вершины опорного многоугольника, определяющего искомую кривую опорной траектории, как B_0, B_1, B_2, B_3 . Пусть известно, что положение вершины B_0 совпадает с положением заднего колеса начального пространственного расположения мобильной платформы (имеет координаты $[0,0]$ относительно введённой системы координат $x_1O_1y_1$), а координаты вершины B_3 определяются заданным пространственным расположением мобильной платформы. Будем считать, что координата x вершины B_3 находится на расстоянии $3R_{min}$ от точки начала движения [119], т.е.:

$$B_{3x} = L_1 + L_2 = 3R_{min}. \quad (3.25)$$

Данное решение основывается на том, что расстояние в $3R_{min}$ является достаточным для выполнения любого манёвра, в том числе полного разворота мобильной платформы, без снижения скорости движения, что обусловлено конструкцией четырехколёсных транспортных средств, имеющих автомобильную кинематическую схему. В свою очередь координату y для вершины B_3 будем определять, используя следующую формулу:

$$B_{3y} = M_y + L_{63}. \quad (3.26)$$

Здесь $L_{63} = const$, расстояние наиболее близких друг к другу точек мобильной платформы и препятствия в момент гипотетического столкновения. [103]

Таким образом, исходное пространство поиска опорной траектории сужается до задачи определения множеств потенциально допустимых координат вершин B_1 и B_2 . Однако в данной постановке задача поиска опорной траектории является неразрешимой, т.к. существует бесконечное количество вариаций расположения вершин B_1 и B_2 , при которых определённая ими кривая будет пригодной для использования в качестве опорной траектории. Поэтому сузим исходное пространство поиска, введя дополнительные ограничения. [120]

Будем считать, что $B_{1y} = B_{0y}$ и $B_{2y} = B_{4y}$, что существенно сокращает количество возможных альтернатив, при этом не теряется практическая ценность полученного решения. А также предположим, что координаты x вершин B_1, B_2 принадлежат интервалу $[0, R_{min}]$, данное допущение основано на том, что изучение кривых, где $B_{1x}, B_{2x} \rightarrow B_{3x}0$, лишено всякого смысла, что показано в публикации [119], этот же факт подтвержден результатами значительного числа вычислительных экспериментов.

Помимо введённых ограничений, будем считать, что пространство поиска координат x для вершин B_1 и B_2 является дискретным с шагом Δx (шаг определяется желаемым количеством шагов d на интервале $[0, R_{min}]$), т.е.

$$B_{1,2xi} = B_{1,2xi-1} + \Delta x, \text{ где } \Delta x = \frac{R_{min}(v)}{d} = const. \quad (3.27)$$

Тогда с учётом сделанных предположений и введённых ограничений исходное пространство поиска будет представлено областью, изображённой на рисунке 3.10 одинарной штриховкой, и может быть задано в виде таблицы 3.1. [118].

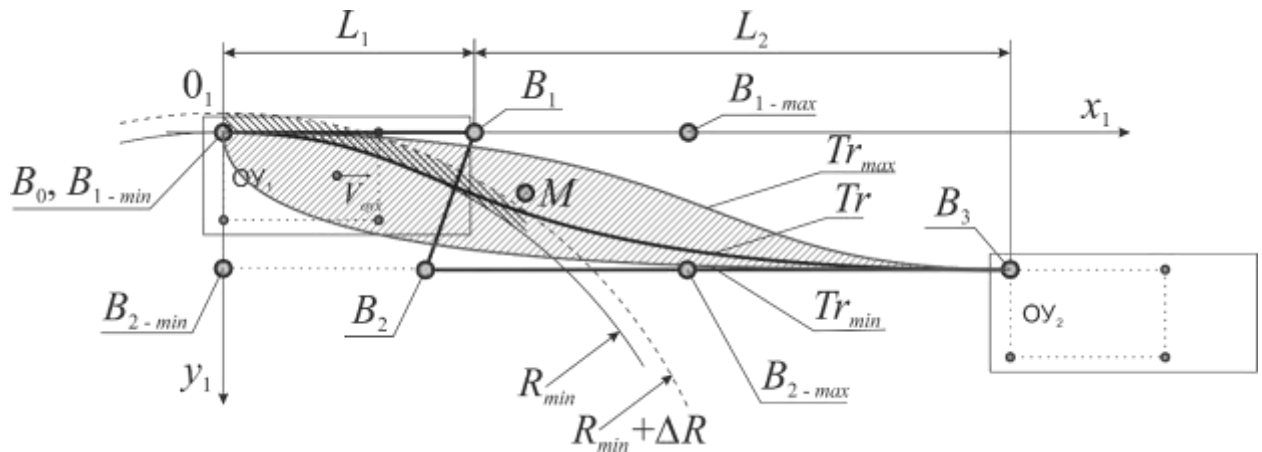


Рисунок 3.10. – Исходная и конечная области поиска кривой опорной траектории. Исходное пространство поиска – одинарная штриховка; Конечное пространство поиска двойная штриховка; Tr_{max} , Tr_{min} , кривые определяющие границы исходного пространства поиска; Tr – кривая опорной траектории. L_1 – активное уклонение; L_2 – релаксация; R_{min} – минимально-допустимый радиус разворота мобильной платформы; $R_{min} + \Delta R$ – буфер безопасности; $B_{1,n}$ – множество допустимых значений координат расположения точки вершины опорного многоугольника.

Таблица 3.1. – Координаты вершин опорного многоугольника

B_0 , м	B_1 , м	B_2 , м	B_3 , м
[0,0]	[0, 0]	[0, $M_y + L_{\phi_3}$]	[$3R_{min}$, $M_y + L_{\phi_3}$]
...
[0,0]	[R_{min} , 0]	[R_{min} , $M_y + L_{\phi_3}$]	[$3R_{min}$, $M_y + L_{\phi_3}$]

Далее, имея полный перечень ограничений, определив шаг Δx и минимальный радиус разворота мобильной платформы R_{min} , можно сформировать исходное пространство поиска кривой опорной траектории, для чего будем использовать итерационный алгоритм [120], блок-схема которого приведена на рисунке 3.11.

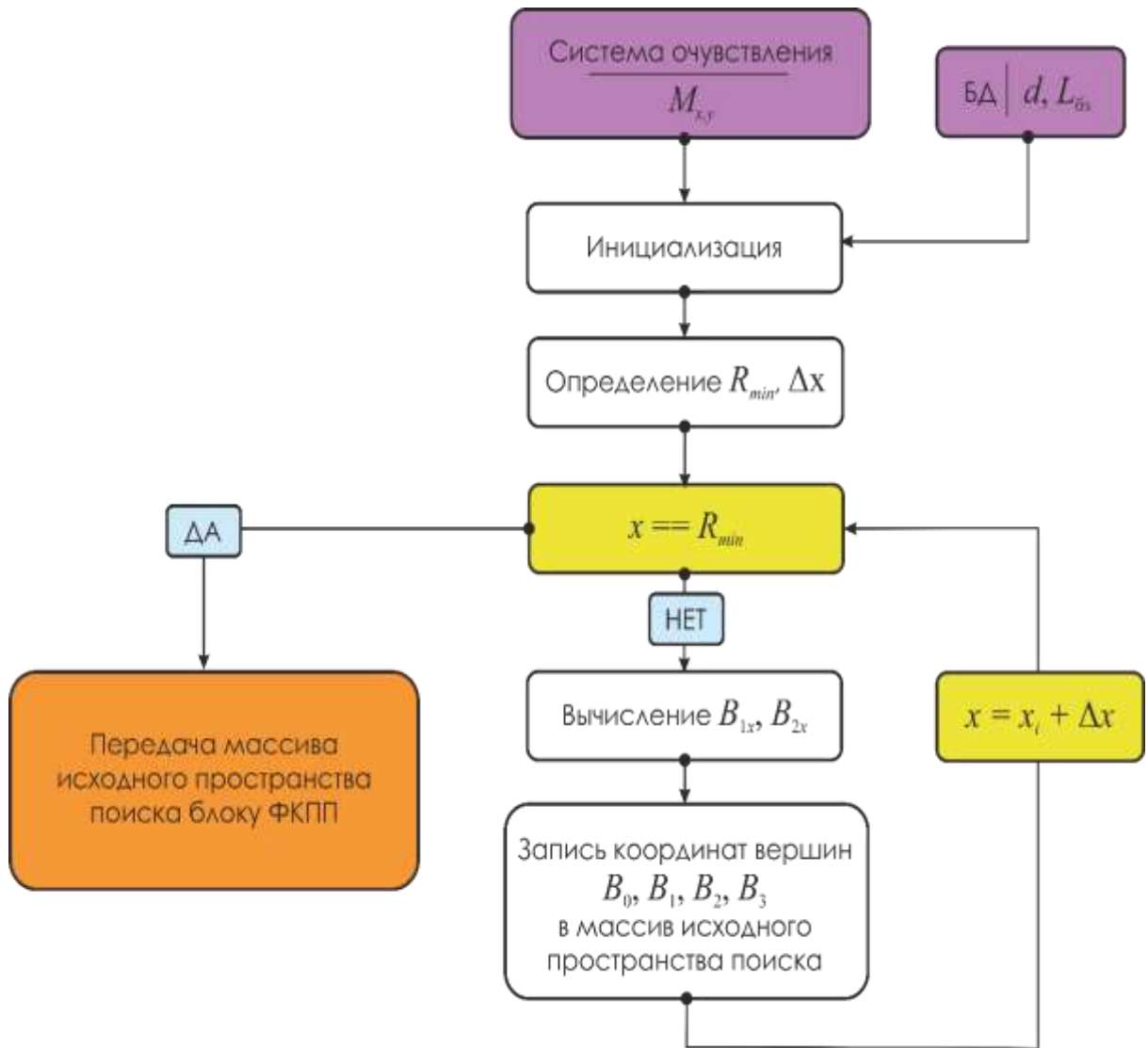


Рисунок 3.11. – Блок-схема алгоритма формирования исходного пространства поиска кривой опорной траектории.

3.5 Формирование конечного пространства поиска

Пусть на некотором интервале L_1 необходимо определить принадлежность некоторой K кривой трубке заданных ограничений. Пусть также известно, что область допустимых значений трубки ограничений ограничена концентрическими окружностями радиусов R_{min} и $R_{min} + \Delta R$. Обойдём интервал L_1 , двигаясь по кривой вдоль оси x с некоторым шагом Δx , много меньшим ΔR (рисунок 3.12).

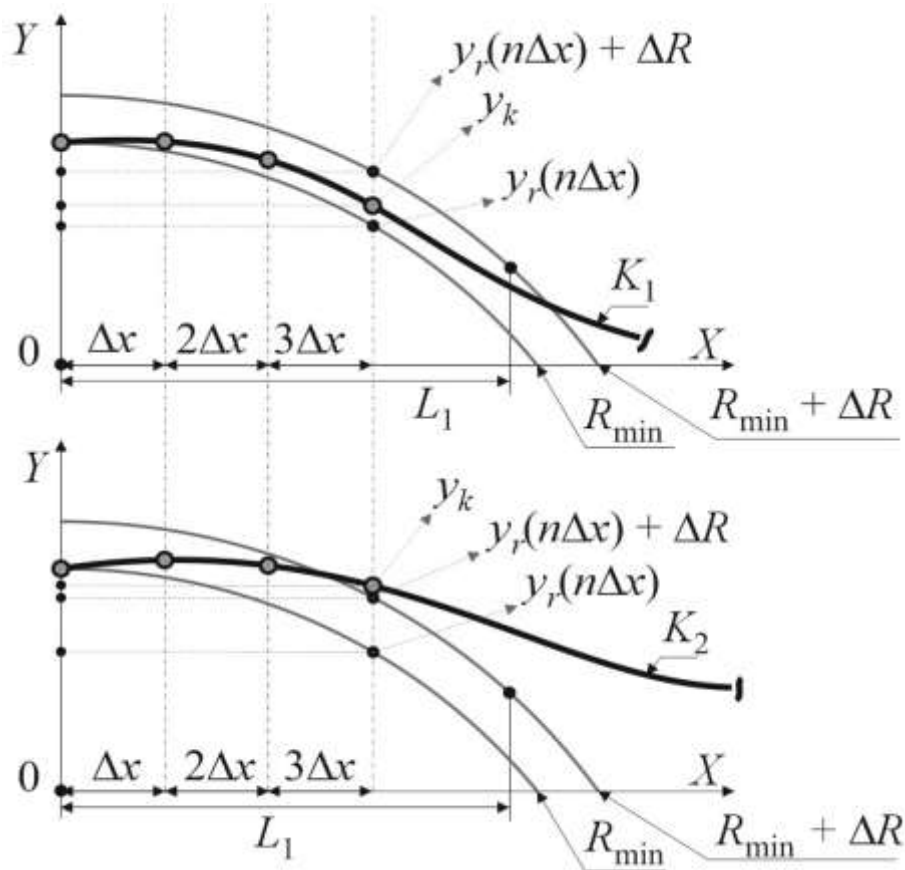


Рисунок 3.12. – Иллюстрация алгоритма выбраковки кривых исходного пространства поиска.

Тогда условие попадания кривой в трубку ограничений можно записать следующим образом:

$$y_r(n\Delta x) + \Delta R, \text{ при } 0 < n < \text{round}\left(\frac{L_1}{\Delta x}\right), \quad (3.28)$$

где $y_r(n\Delta x)$ – соответствующая координата y ограничивающей окружности, а n – номер шага по кривой. Блок-схема, иллюстрирующая работу алгоритма выбраковки кривых исходного пространства поиска, показана на рисунке 3.13.

Таким образом, на выходе блока ФКПП, реализующего алгоритм выбраковки кривых, мы получаем массив кривых конечного пространства поиска, каждая из которых может быть использована в качестве опорной траектории для модуля ФУС. Следующим шагом необходимо произвести выбор единственной – наилучшей кривой из определённого конечного множества альтернатив.

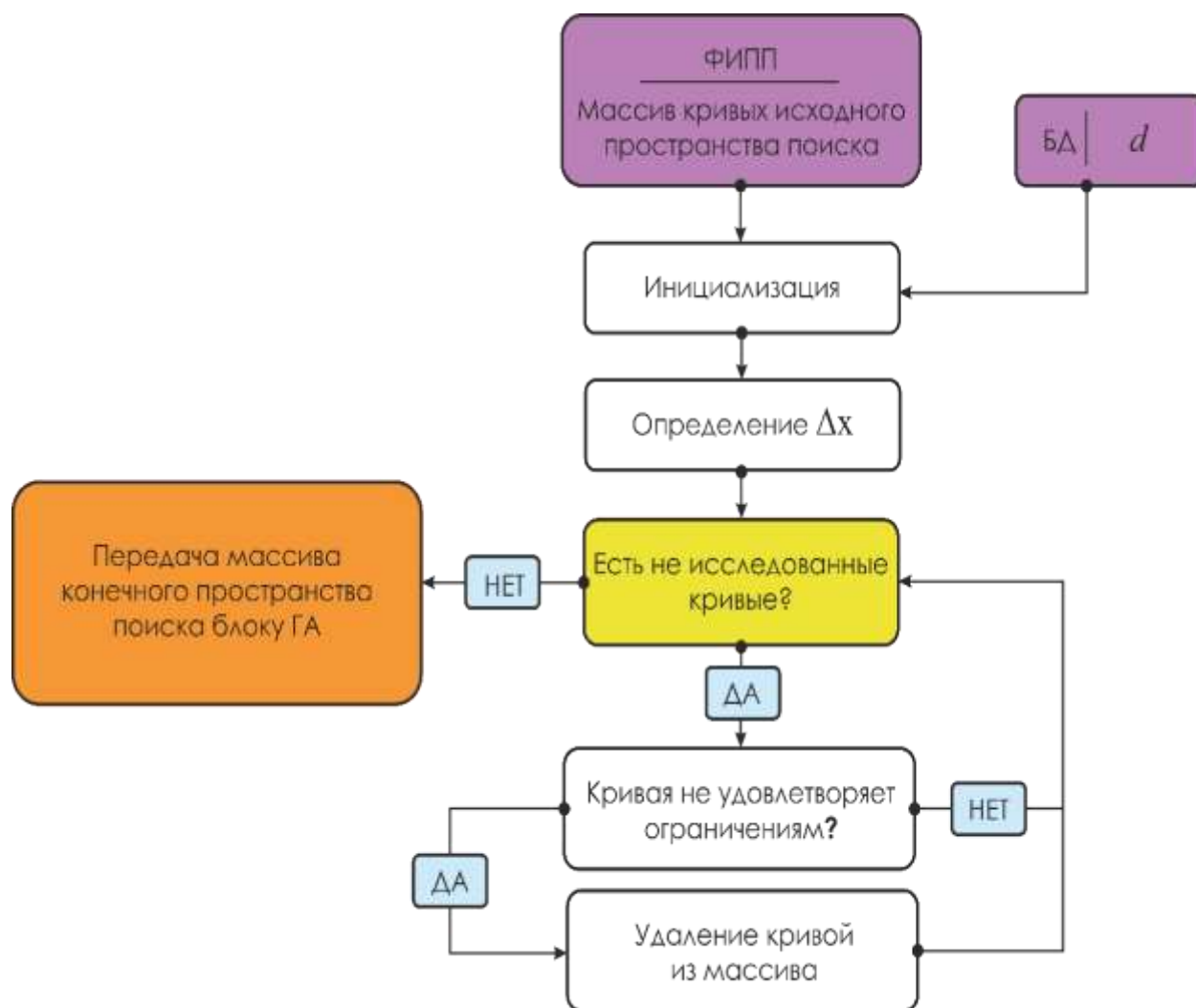


Рисунок 3.13 – Алгоритм программы выбраковки кривых исходного пространства поиска.

3.6 Выбор кривой опорной траектории из конечного множества кривых, удовлетворяющих задаче перемещения

Справедливости ради следует заметить, что каждая кривая, принадлежащая конечному пространству поиска, полученному на предыдущем этапе, во-первых, полностью удовлетворяет условию поставленной задачи перемещения, во-вторых, позволяет совершить манёвр безаварийно, т.е. исключает столкновение с

препятствием. Поэтому при решении задач, связанных с автоматизацией транспортировки грузов по территории логистического центра, блок ГА можно исключить из модуля генератора траектории, а в качестве опорной траектории может быть выбрана первая или любая иная (например, случайная) кривая из массива конечного пространства поиска. Другими словами, применение генетического алгоритма на конечном пространстве поиска, в рамках решаемой задачи, представляет более теоретический, нежели практический, интерес. Однако, практическая ценность включения блока ГА в структуру модуля ГТ возрастает с увеличением скорости движения колёсного робота в рабочей зоне, т.к. использование метода эволюционного поиска позволяет выбрать единственную наилучшую опорной кривую. Здесь под наилучшей кривой будем понимать такую кривую, при использовании которой в качестве опорной траектории, время движения робота на участке активной фазы уклонения является минимальным.

Основанием для выбора генетического алгоритма в качестве способа определения наилучшей кривой из сформированного пространства поиска послужил тот факт, что эволюционные методы имеют широкое распространение при решении задач, связанных с поиском в сложном пространстве решений большой размерности, что показано в работах [121, 122], причём их эффективность является доказанной и многократно подтверждённой. Помимо этого, генетические алгоритмы обладают целым рядом достоинств – это, во-первых, отсутствие ограничений на вид целевой функции, во-вторых, очевидность базовых принципов и схемы вычислений и, в-третьих, простота их машинной реализации [123].

Из соображений лаконичности изложения не будем подробно рассматривать принципы эволюционного поиска, т.к. существует достаточное количество работ, где данная тема хорошо освещена, укажем только, что в рамках настоящего исследования используется терминология генетических алгоритмов, приведённая в источнике [123], что необходимо, т.к. в разных трудах существуют терминологические расхождения.

Дополнительно отметим, что в процессе выполнения данной работы в качестве реализации генетического алгоритма использовался классический генетический алгоритм [126], блок-схема которого приведена на рисунке 3.14.

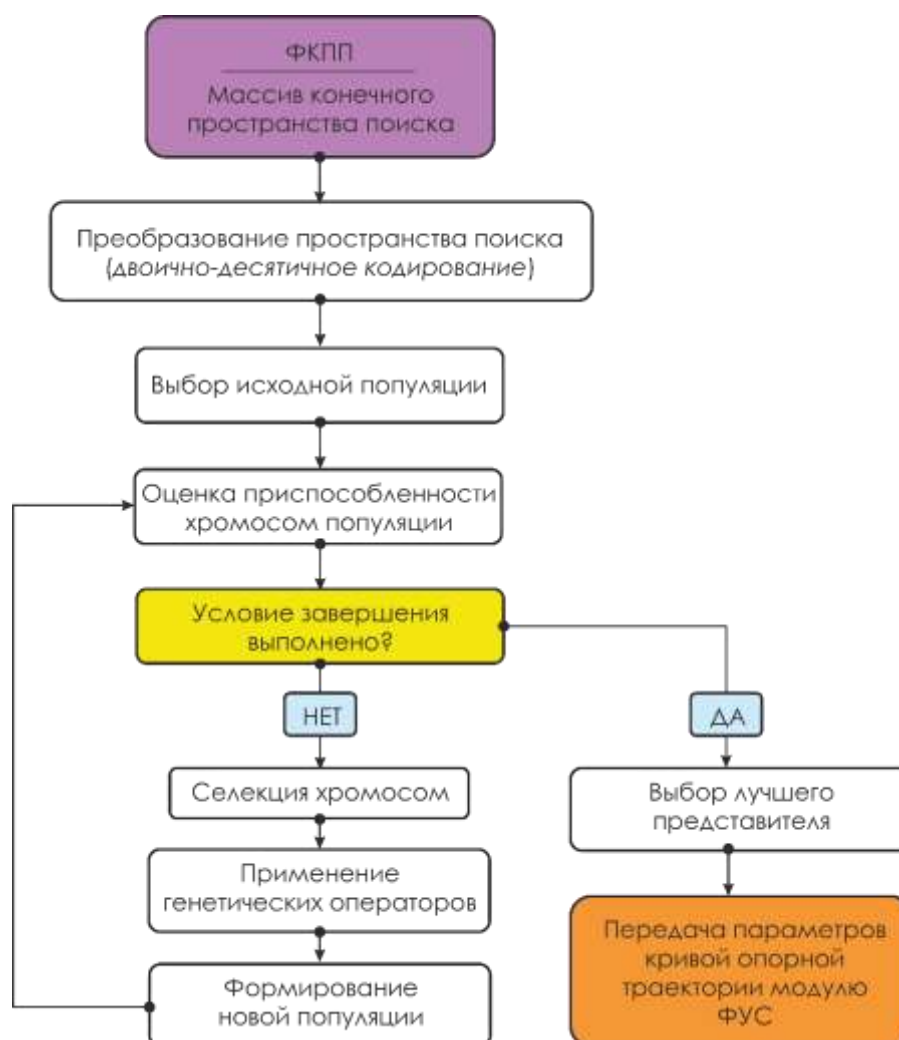


Рисунок 3.14. – Блок-схема использованного генетического алгоритма

Селекция хромосом осуществлялась в соответствии с получившим широкое распространение принципом «колеса рулетки» [125], в качестве целевой функции использовалась выражение (3.16). За условия остановки поиска были приняты, во-первых, отсутствие улучшения среднего значения целевой функции для текущего поколения по сравнению с предыдущим на дистанции в n_d итераций, во-вторых, превышение максимального числа итераций $n_{\max} = const$. Операторы скрещивания и мутации были реализованы стандартным способом.

Также отметим, что применение генетического алгоритма для поиска наилучшей альтернативы на конечном множестве поиска потребовало замены переменных, т.к. основными требованиями к множеству альтернатив, на котором применяется генетический алгоритм, является, во-первых, конечность исследуемого множества, во-вторых, его непрерывность. Очевидно, что полученное на предыдущем этапе пространство поиска кривой опорной траектории удовлетворяет только требованию конечности, но не является непрерывным. Поэтому каждая кривая, принадлежащая пространству поиска, была пронумерована от 0 до $N-1$, где N равно числу кривых, входящих в пространство поиска, а затем её номер подвергался двоично-десятичному кодированию. Таким образом, удалось привести имеющееся пространство поиска к непрерывному конечному кодовому пространству, на котором можно использовать генетический алгоритм [127].

Дополнительно заметим, что вопрос, касающийся оправданности применения генетического алгоритма как такового в задачах выбора наилучшего маршрута из множества альтернатив, имеет право на существование. Так как к поиску лучшей альтернативы можно подойти путём решения экстремальной задачи в классической постановке, однако в случае значительного множества альтернативных маршрутов $N > 300$, генетические алгоритмы дают уверенно лучший результат в отношении времени затраченного на поиск решения, что показано в работах [128, 129]. В рамках настоящего исследования выводы, сделанные авторами указанных работ, были проверены и подтверждены в рамках текущей задачи, результаты вынесены в Приложение Б.

3.7 Выводы по главе 3

- Разработанная структурная схема реализации модуля бортовой центральной вычислительной системы (ЦВС) полностью удовлетворяет выбранной стратегии управления движением колёсного робота.

- На основании проведённого анализа особенности конструкции четырёхколёсных транспортных средств, имеющих автомобильную кинематическую схему, установлено наличие жёсткой механической связи между траекторией движения переднего и заднего колеса, расположенных на одной продольной оси мобильной платформы.

- Предложенная модель представления мобильной платформы как объекта управления учитывает существующие механические связи между траекторией движения управляющих и свободно катящихся колёс и является удобной альтернативой для реализации алгоритмов построения маршрутов колёсного робота.

- Выбор кривых Безье третьего порядка в качестве опорных траекторий для построения маршрута колёсного робота, обеспечивает простоту машинной обработки маршрутов, и как следствие снижение технических требований к аппаратной реализации системы управления роботом.

- Разработанный алгоритм, позволяющий колёсному роботу осуществлять обработку произвольных криволинейных маршрутов, обеспечивает возможность точного определения пространственного расположения мобильной платформы во время исполнения задания.

- Разработанное алгоритмическое обеспечение модуля генератора опорных траекторий, позволяет формировать маршруты, исключая столкновения со статическими или движущимися препятствиями, внезапно возникшими на пути следования робота.

Глава 4. Экспериментальное исследование разработанного алгоритмического обеспечения системы управления колёсным роботом

В рамках настоящего исследования разработанная группа алгоритмов была подвергнута серии испытаний, которые представляют собой вычислительные и полунатурные эксперименты, целью которых было подтверждение практической ценности полученных теоретических решений. Результаты проведённых исследований, перечень полученных данных, а также выводы, сделанные на основе проведённых тестов, приведены ниже.

4.1 Исполнение задания, парковка робота в ограниченном пространстве

Суть эксперимента заключалась в поиске набора управляющих воздействий при выполнении манёвра парковки колёсного робота в ограниченном пространстве с последующим анализом аппроксимации фактически полученной траектории движения переднего колеса. Данный манёвр является важной частью технологического процесса автоматизированной транспортировки грузов по территории логистического центра, т.к. выполняется каждый раз при осуществлении подзарядки бортовых аккумуляторных батарей робота.

Предполагалась, что схема маневрирования аналогична манёвру «параллельная парковка» для легкового автомобиля [49], рисунок 4.1, здесь парковочный «карман» ограничен зарядной платформой.

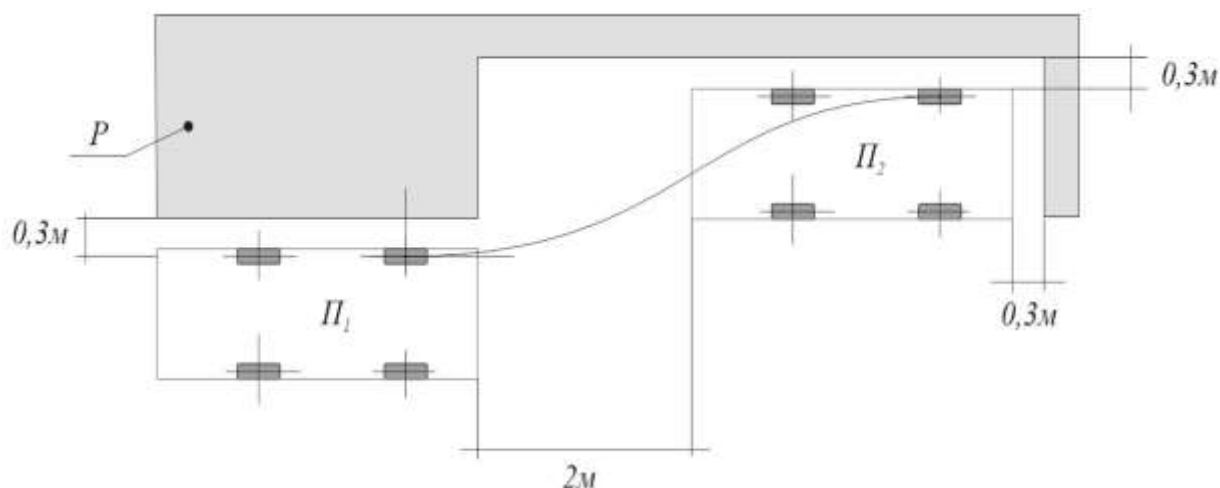


Рисунок 4.1 – Схема манёвра «Парковка в ограниченном пространстве». Π_1 – начальное пространственное расположение, Π_2 – заданное пространственное расположение.

В качестве расчётных параметров мобильной платформы здесь и далее использованы габаритные размеры транспортёра типа ЕТ1 производства БАЛКАН-КАР РЕКОРД АО [102]. Параметры опорной траектории были определены на основании следа заднего колеса, полученного при выполнении маневрирования аналогичным транспортным средством под управлением человека-оператора, и сведены в таблицу 4.1.

Таблица 4.1 – Параметры опорной траектории

Координаты вершин опорного многоугольника, манёвра «Парковка в ограниченном пространстве»			
B_0 , м	B_1 , м	B_2 , м	B_3 , м
[0; 0]	[2.68; 0]	[2.68; 1.26]	[5.12; 1.26]

Шаг по кривой опорной траектории $\Delta t = 0,05$. Скорость движения мобильной платформы является постоянной и составляет $0,638$ м/с ($2,3$ км/ч), движение осуществляется задним ходом.

Целью эксперимента заключалась в том, чтобы оценить расхождение между эталонной кривой (на рисунке 4.2 обозначена – B) переднего колеса, полученной с реального транспортёра, и её аппроксимацией, полученной посредством сделанных вычислений.

В качестве допустимого отклонения от продольной оси следа эталонной траектории было принято значение $\Delta\varepsilon = 0,05\text{м}$, что обусловлено шириной следа колеса рассматриваемого транспортёра.

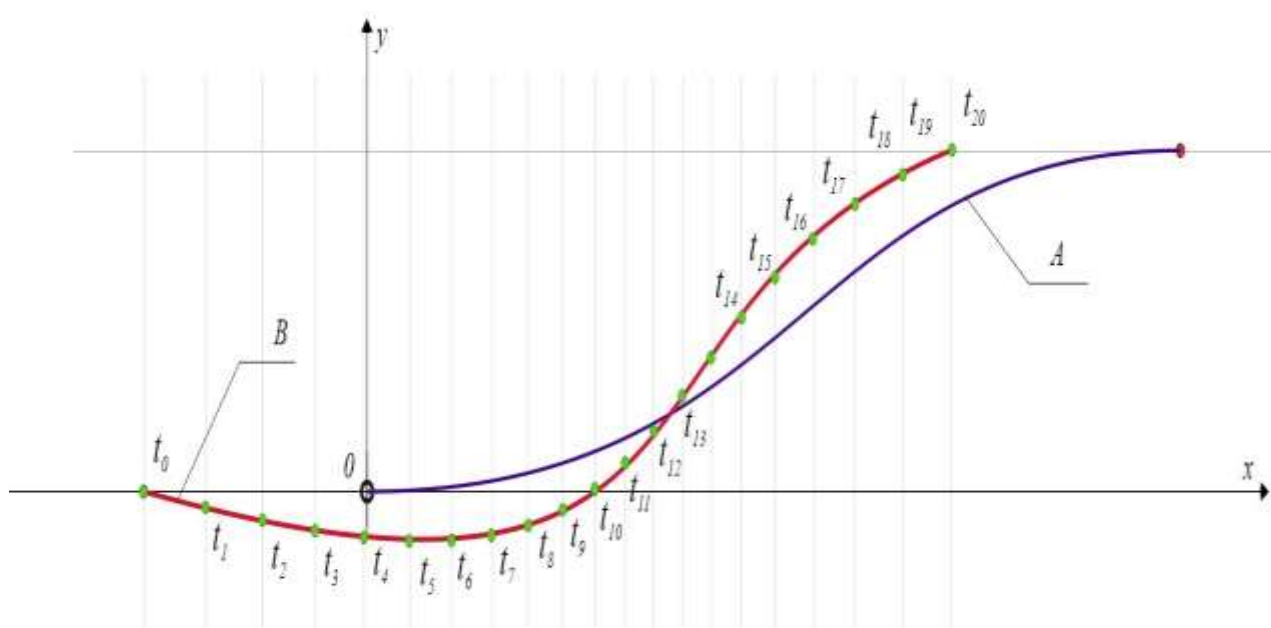


Рисунок 4.2 – Аппроксимация траектории переднего колеса для манёвра «парковка в ограниченном пространстве». A – Кривая опорной траектории; B – эталонная кривая траектории движения переднего колеса; $t_1 \dots t_{20}$ – вычисленные координаты на траектории переднего колеса

В результате проведённого вычислительного эксперимента были получены следующие данные. Максимальное отклонение от продольной оси следа эталонной траектории составило $0,01471\text{м}$, полный перечень отклонений приведён на рисунке 4.3.

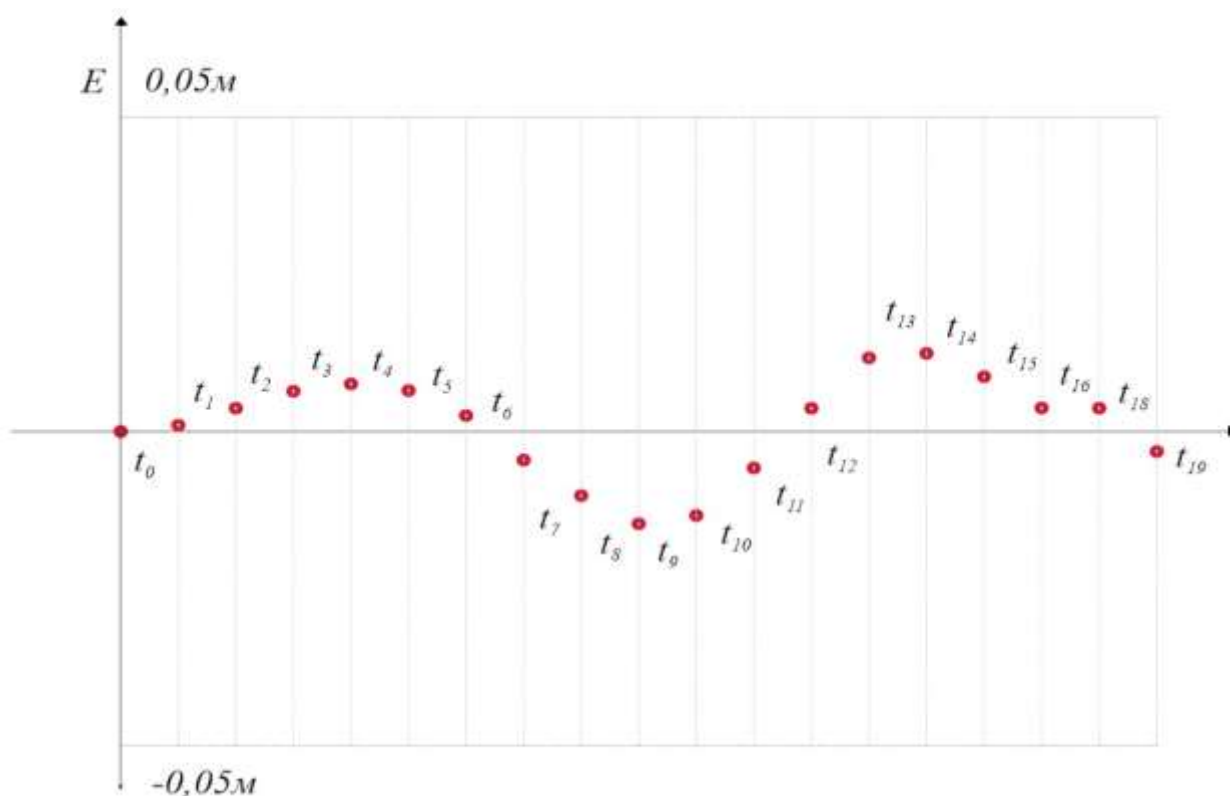


Рисунок 4.3 – Отклонения от продольной оси следа заранее известной траектории, $\Delta t = 0,05$

Для контрольного испытания при $t = 0,05$ рассчитаны дисперсия и математическое ожидание, составившие: $D = 5.5988999E-05\text{м}^2$, $M = -0.001009\text{м}$, соответственно.

4.2 Исполнение задания, траектория ухода от столкновения с препятствием

В рамках эксперимента заранее определённая модулем ГТ опорная траектория передавалась модулю ФУС, которым был произведён расчёт задания, необходимого для отработки опорной траектории. Входные параметры опорной траектории, а также константы, использованные в процессе вычислений, сведены в таблицу 4.2.

Таблица 4.2 – Входные параметры, использованные в процессе вычислений

Координаты вершин опорного многоугольника исследуемой кривой				Шаг, м	Длина базы робота, м	Точность округления
B_0	B_1	B_2	B_3	Δt	$L_{бд}$	p
[0; 20]	[20; 20]	[10.6; 0]	[60; 0]	0,025	5	10^{-3}

При проведении эксперимента считалось, что мобильная платформа движется по траектории со скоростью 2,777м/с (10км/ч). Пошаговая трассировка процесса определения управляющих воздействий приведена в таблице 4.3, где № – номер шага алгоритма; t_i – значение параметра t ; $P_{x,y}(t_i)$ – координаты точки на опорной траектории; $P'_{x,y}(t_i)$ – координаты точки на касательной к кривой опорной траектории; $y_{ki} = f(x)$ – уравнение касательной; L – длина отрезка аппроксимации; $O_{x,y}(t_i)$ – точка пересечения окружности радиуса R_6 и касательной к кривой (точка, определяющая положение переднего колеса); α_i – угол отклонения переднего колеса от продольной оси; β_i – управляющее воздействие на текущем шаге.

На рисунке 4.4 показана исходная кривая и кусочно-линейная аппроксимация для переднего колеса мобильной платформы. Кривая выстроена по результатам таблицы 4.3 в программе CorelDrawX5 в масштабе 10:1. Рисунок 4.5 иллюстрирует имеющуюся жёсткую связь между положениями переднего и заднего колеса для каждого отдельно взятого тела цикла. Отклонение от заранее известной кривой в каждом отдельно взятом шаге цикла алгоритма не превышает максимально допустимое $\Delta \varepsilon = 0,05$ м, максимальное зафиксированное отклонение 0,02164м.

Таблица 4.3 – Пошаговая трассировка процесса определения управляющих воздействий

№ шага	t_i	$P_{x,y}(t_i)$	$P'_{x,y}(t_i)$	$y_{ki} = f(x)$	L	$O_{x,y}(t_i)$	α_i	β_i
1	0	0,000; 20,000	60,000; 0,000	$y(x) = 20$	0	5,000; 20,000	0	0
2	0,025	1,446; 19,963	55,755; -2,925	$y(x) = 0.052x + 20.038$	1,447	6,439; 19,701	0,079	0,079
3	0,05	2,791; 19,855	51,842; -5,700	$y(x) = 0.11x + 20.162$	1,349	7,761; 19,308	0,139	0,06
4	0,075	4,041; 19,679	48,258; -8,325	$y(x) = 0.173x + 20.378$	1,263	8,968; 18,829	0,202	0,063
5	0,1	5,206; 19,440	45,006; -10,80	$y(x) = 0.24x + 20.689$	1,189	0,344; 20,607	0,268	0,066
6	0,125	6,294; 19,141	42,084; -13,125	$y(x) = 0.312x + 21.104$	1,128	1,521; 20,629	0,336	0,068
7	0,15	7,313; 18,785	39,494; -15,300	$y(x) = 0.387x + 21.615$	1,079	2,651; 20,591	0,403	0,067
8	0,175	8,272; 18,377	37,233; -17,325	$y(x) = 0.465x + 22.223$	1,042	3,738; 20,486	0,468	0,065
9	0,2	9,178; 17,920	35,304; -19,200	$y(x) = 0.544x + 22.913$	1,015	4,785; 20,309	0,529	0,061
10	0,225	10,040; 17,418	33,705; -20,925	$y(x) = 0.621x + 23.653$	0,997	5,792; 20,055	0,584	0,055
11	0,25	10,866; 16,875	32,438; -22,500	$y(x) = 0.694x + 24.416$	0,989	6,757; 19,725	0,631	0,047
12	0,275	11,664; 16,294	31,500; -23,925	$y(x) = 0.76x + 25.159$	0,987	7,682; 19,319	0,67	0,039
13	0,3	12,443; 15,680	30,894; -25,200	$y(x) = 0.816x + 25.834$	0,992	8,569; 18,840	0,701	0,031
14	0,325	13,212; 15,036	30,618; -26,325	$y(x) = 0.86x + 26.398$	1,003	9,420; 18,296	0,722	0,021
15	0,35	13,977; 14,365	30,674; -27,300	$y(x) = 0.89x + 26.805$	1,018	10,242; 17,689	0,735	0,013
16	0,375	14,748; 13,672	31,059; -28,125	$y(x) = 0.906x + 27.034$	1,037	11,042; 17,028	0,739	0,004
17	0,4	15,533; 12,960	31,776; -28,800	$y(x) = 0.906x + 27.033$	1,06	11,828; 16,318	0,736	-0,003
18	0,425	16,340; 12,233	32,823; -29,325	$y(x) = 0.893x + 26.824$	1,086	12,611; 15,564	0,725	-0,011
19	0,45	17,177; 11,495	34,202; -29,700	$y(x) = 0.868x + 26.404$	1,116	13,401; 14,774	0,707	-0,018
20	0,475	18,052; 10,749	35,910; -29,925	$y(x) = 0.833x + 25.787$	1,15	14,211; 13,951	0,684	-0,023

Таблица 4.3 - продолжение

№ шага	t_i	$P_{x,y}(t_i)$	$P'_{x,y}(t_i)$	$y_{ki} = f(x)$	L	$O_{x,y}(t_i)$	α_i	β_i
21	0,5	18,975; 10,000	37,950; -30,000	$y(x) = 0.791x + 25.009$	1,189	15,053; 13,100	0,656	-0,028
22	0,525	19,953; 9,251	40,320; -29,925	$y(x) = 0.742x + 24.056$	1,232	15,938; 12,230	0,623	-0,033
23	0,55	20,994; 8,505	43,022; -29,700	$y(x) = 0.69x + 22.991$	1,281	16,879; 11,346	0,587	-0,036
24	0,575	22,107; 7,767	46,053; -29,325	$y(x) = 0.637x + 21.849$	1,335	17,889; 10,452	0,548	-0,039
25	0,6	23,299; 7,040	49,416; -28,800	$y(x) = 0.583x + 20.623$	1,397	18,979; 9,558	0,508	-0,04
26	0,625	24,580; 6,328	53,109; -28,125	$y(x) = 0.53x + 19.356$	1,465	20,161; 8,668	0,467	-0,041
27	0,65	25,957; 5,635	57,134; -27,300	$y(x) = 0.478x + 18.043$	1,542	21,446; 7,791	0,425	-0,042
28	0,675	27,440; 4,964	61,488; -26,325	$y(x) = 0.428x + 16.708$	1,627	22,843; 6,932	0,384	-0,041
29	0,7	29,035; 4,320	66,174; -25,200	$y(x) = 0.381x + 15.382$	1,72	24,362; 6,099	0,344	-0,04
30	0,725	30,751; 3,706	71,190; -23,925	$y(x) = 0.336x + 14.038$	1,823	26,011; 5,299	0,305	-0,039
31	0,75	32,597; 3,125	76,538; -22,500	$y(x) = 0.294x + 12.708$	1,935	27,800; 4,535	0,267	-0,038
32	0,775	34,581; 2,582	82,215; -20,925	$y(x) = -0.255x + 11.4$	2,057	29,735; 3,815	0,231	-0,036
33	0,8	36,710; 2,080	88,224; -19,200	$y(x) = 0.218x + 10.083$	2,188	31,825; 3,143	0,197	-0,034
34	0,825	38,995; 1,623	94,563; -17,325	$y(x) = -0.183x + 8.759$	2,329	34,076; 2,524	0,165	-0,032
35	0,85	41,441; 1,215	101,234; -15,300	$y(x) = -0.151x + 7.473$	2,481	36,497; 1,962	0,135	-0,03
36	0,875	44,059; 0,859	108,234; -13,125	$y(x) = -0.121x + 6.191$	2,642	39,095; 1,461	0,106	-0,029
37	0,9	46,856; 0,560	115,566; -10,800	$y(x) = -0.093x + 4.918$	2,813	41,877; 1,025	0,08	-0,026
38	0,925	49,840; 0,321	123,228; -8,325	$y(x) = -0.068x + 3.71$	2,994	44,851; 0,658	0,055	-0,025
39	0,95	53,020; 0,145	131,222; -5,700	$y(x) = -0.043x + 2.425$	3,185	48,025; 0,362	0,032	-0,023
40	0,975	56,404; 0,037	139,545; -2,925	$y(x) = -0.021x + 1.221$	3,386	51,405; 0,142	0,01	-0,022

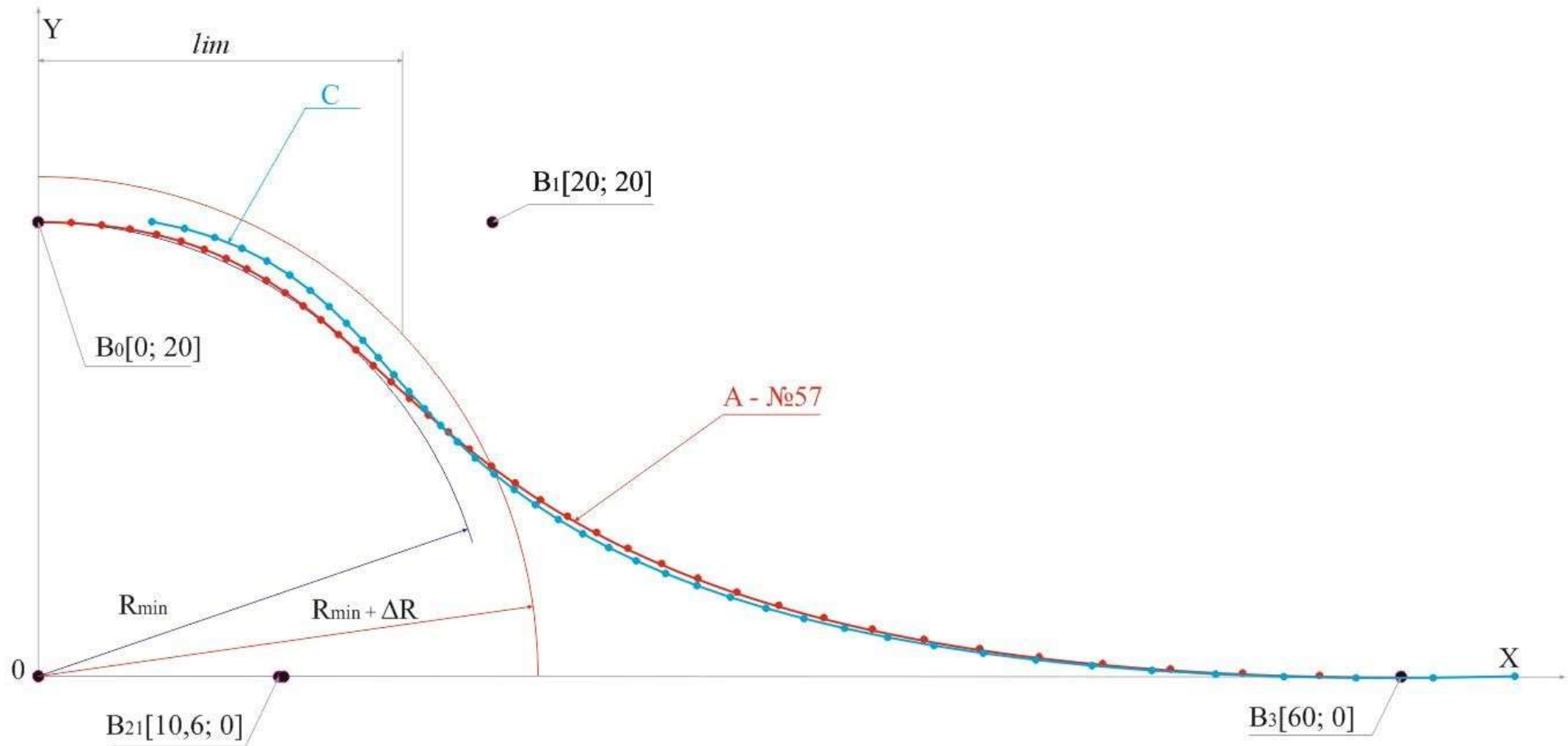


Рисунок 4.4 – Кусочно-линейные аппроксимации траектории заднего и переднего колеса. C – кусочно-линейная аппроксимация траектории переднего колеса мобильной платформы; A – опорная траектория переданная модулю ФУС;

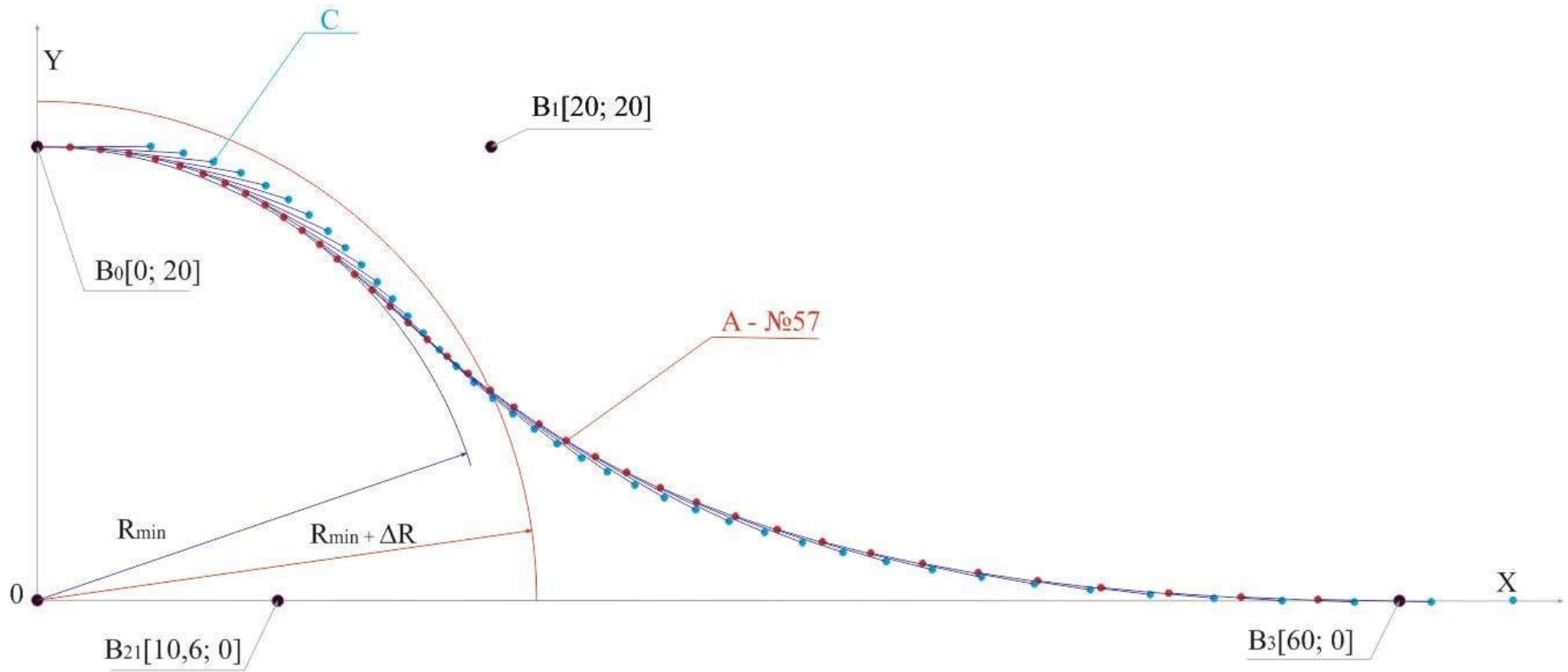


Рисунок 4.5 – Геометрические связи между траекторией переднего и заднего колеса;

4.3 Поиск опорной траектории, уход от столкновения с препятствием

Рассмотрим численный пример процесса поиска кривой опорной траектории.

Так как мобильная платформа имеет автомобильную кинематическую схему, то выбор схемы маневрирования осуществлялся из перечня типовых манёвров, характерных для легковых автомобилей с одной парой управляющих колёс. В частности, в качестве манёвра уклонения от столкновения с внезапно возникшим препятствием был выбран манёвр «переставка», предусмотренный европейским стандартом DIN [131]. Схема манёвра «переставка» изображена на рисунке 4.6.

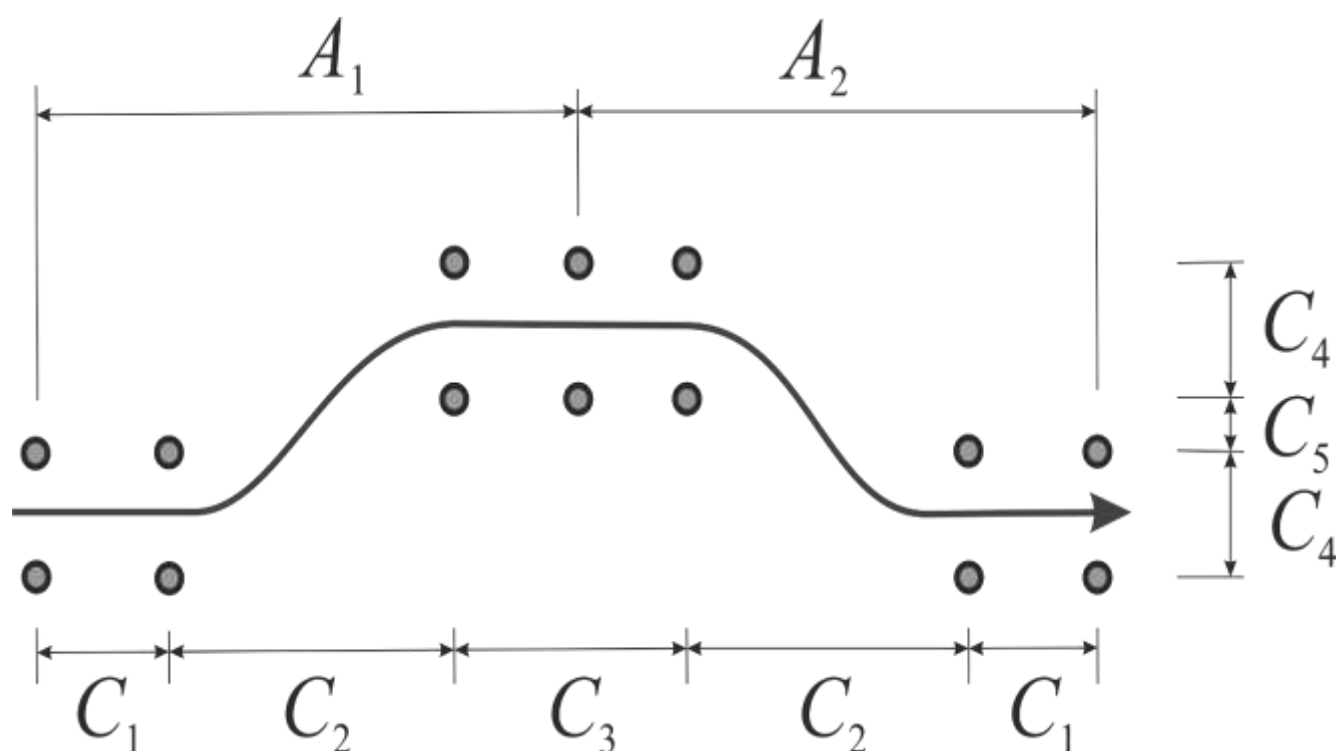


Рисунок 4.6 – Схематичное представление манёвра типа «переставка», соответствующая стандарту DIN. A_1 – фаза активного уклонения; A_2 – фаза релаксации; C_1 - C_5 – параметры манёвра, определяются габаритами транспортного средства

В силу особенностей типа кривых, выбранных в качестве опорных траекторий, мы не имеем возможности спланировать опорную траекторию для манёвра такого типа целиком, т.е. маневрирование должно осуществляться в два этапа. Сначала необходимо уйти от столкновения (рисунок 4.6, участок A_1), затем выполнить возврат к прежней траектории (участок A_2). Из соображений лаконичности покажем трассировки эксперимента только для участка A_1 . Специально отметим, на момент начала поиска нам известно, что наилучшая кривая для данных условий имеет следующие координаты вершин опорного многоугольника: $B_0[0;20]$, $B_1[17.8;20]$, $B_2[16.6;0]$, $B_3[60;0]$. Данную кривую по условию эксперимента будем считать эталонной.

Входные данные, использованные в процессе поиска кривой опорной траектории, приведены в таблице 4.4.

Таблица 4.4 – Поиск опорной траектории, входные данные

B_{1x}, B_{2x}, M	d	p	V м/с	R_{min}	ΔR , м
[0; 20]	100	10^{-3}	2,777	20	2

Здесь B_{1x}, B_{2x} – интервал, на котором осуществляется поиск координат x точек опорного многоугольника опорной кривой траектории; d – количество шагов алгоритма ($\Delta x = L_1/d = 0,2$); p – точность машинных вычислений; V м/с – скорость движения робота; R_{min} – минимальный радиус разворота мобильной платформы; ΔR – радиус буферной зоны.

Применив алгоритм формирования исходного пространства поиска, получим множество всех кривых, удовлетворяющих заданным начальным условиям. Координаты вершин опорных многоугольников (B_0, B_1, B_2, B_3) для кривых, ограничивающих исходное пространство поиска, приведены в таблице 4.5 (здесь № – номер кривой в исходном пространстве поиска).

Таблица 4.5 – Исходное пространство поиска

№	B_0	B_1	B_2	B_3
1	[0, 0]	[0, 0]	[0, 20]	[60, 20]
...
10201	[0, 0]	[20, 0]	[20, 20]	[60, 20]

На сформированном исходном пространстве поиска проведём операцию выбраковки кривых, не удовлетворяющих условию (3.28), посредством алгоритма формирования конечного пространства поиска. Трассировка процесса выбраковки кривой показана на примере двух кривых исходного пространства поиска и приведена в таблицах 4.6 и 4.7. Для наглядности кривые выбирались таким образом, чтобы одна из них была отбракована, а вторая попала в конечное пространство поиска. Кривые, ограничивающие полученное в результате выбраковки конечное пространство поиска, приведены в таблице 4.8.

Таблица 4.6. – Выбраковка кривой, кривая №5027 (отбракована)

n	P_{nx}	P_{ny}	Y_{nc}	Y_{nmax}
0	0	20	20	22
1	0,5835408	21,992	19,991	21,991

Таблица 4.7 – Выбраковка кривой, кривая №9563 (включена в КПП)

n	P_{nx}	P_{ny}	Y_{nc}	Y_{nmax}
0	0	20	20	22
1	1,10004	19,97632	19,97	21,97
2	2,14656	19,90656	19,884	19,90656
3	3,14316	19,79264	19,751	19,79264
4	4,09344	19,63648	19,577	19,63648
5	5,001	19,44	19,365	19,44
6	5,86944	19,20512	19,119	19,20512
7	6,70236	18,93376	18,844	18,93376

Таблица 4.7 – продолжение

n	P_{nx}	P_{ny}	Y_{nc}	Y_{nmax}
8	7,50336	18,62784	18,539	18,62784
9	8,27604	18,28928	18,207	18,28928
10	9,024	17,92	17,848	17,92
11	9,75084	17,52192	17,462	17,52192
12	10,46016	17,09696	17,047	17,09696
13	11,15556	16,64704	16,6	16,64704
14	11,84064	16,17408	16,118	16,17408
15	12,519	15,68	15,597	15,68
16	13,19424	15,16672	15,03	15,16672
17	13,86996	14,63616	14,409	14,63616
18	14,54976	14,09024	13,722	14,09024
19	15,23724	13,53088	12,955	13,53088
20	15,936	12,96	12,085	12,96
21	16,64964	12,37952	11,081	12,37952
22	17,38176	11,79136	9,893	11,79136

Здесь n – номер итерации алгоритма; P_{nx} – координата x точки исследуемой кривой на n -ом шаге алгоритма; P_{ny} – координата y точки исследуемой кривой на n -ом шаге алгоритма; Y_{nc} – координата y контрольной точки на окружности; Y_{nmax} – максимально допустимое значение координаты y точки кривой на текущей итерации.

Таблица 4.8 – Конечное пространство поиска

№ _к	№ _{ГА}	№ _{СD}	B_0	B_1	B_2	B_3
9072	0	0000000	[0;20]	[17.8;20]	[16.4;0]	[60;0]
9073	1	0000001	[0;20]	[17.8;20]	[16.6;0]	[60;0]
9172	2	0000010	[0;20]	[18;20]	[16.2;0]	[60;0]
9271	3	0000011	[0;20]	[18.2;20]	[15.8;0]	[60;0]

Таблица 4.8 – продолжение

№к	№ГА	№СД	В ₀	В ₁	В ₂	В ₃
9272	4	0000100	[0;20]	[18.2;20]	[16;0]	[60;0]
9371	5	0000101	[0;20]	[18.4;20]	[15.6;0]	[60;0]
9372	6	0000110	[0;20]	[18.4;20]	[15.8;0]	[60;0]
9464	7	0000111	[0;20]	[18.6;20]	[14;0]	[60;0]
9465	8	0001000	[0;20]	[18.6;20]	[14.2;0]	[60;0]
9471	9	0001001	[0;20]	[18.6;20]	[15.4;0]	[60;0]
9562	10	0001010	[0;20]	[18.8;20]	[13.4;0]	[60;0]
9563	11	0001011	[0;20]	[18.8;20]	[13.6;0]	[60;0]
9564	12	0001100	[0;20]	[18.8;20]	[13.8;0]	[60;0]
9570	13	0001101	[0;20]	[18.8;20]	[15;0]	[60;0]
9571	14	0001110	[0;20]	[18.8;20]	[15.2;0]	[60;0]
9661	15	0001111	[0;20]	[19;20]	[13;0]	[60;0]
9662	16	0010000	[0;20]	[19;20]	[13.2;0]	[60;0]
9663	17	0010001	[0;20]	[19;20]	[13.4;0]	[60;0]
9664	18	0010010	[0;20]	[19;20]	[13.6;0]	[60;0]
9670	19	0010011	[0;20]	[19;20]	[14.8;0]	[60;0]
9759	20	0010100	[0;20]	[19.2;20]	[12.4;0]	[60;0]
9760	21	0010101	[0;20]	[19.2;20]	[12.6;0]	[60;0]
9761	22	0010110	[0;20]	[19.2;20]	[12.8;0]	[60;0]
9762	23	0010111	[0;20]	[19.2;20]	[13;0]	[60;0]
9763	24	0011000	[0;20]	[19.2;20]	[13.2;0]	[60;0]
9764	25	0011001	[0;20]	[19.2;20]	[13.4;0]	[60;0]
9769	26	0011010	[0;20]	[19.2;20]	[14.4;0]	[60;0]

Таблица 4.8 – продолжение

№к	№ГА	№СД	В ₀	В ₁	В ₂	В ₃
9770	27	0011011	[0;20]	[19.2;20]	[14.6;0]	[60;0]
9858	28	0011100	[0;20]	[19.4;20]	[12;0]	[60;0]
9859	29	0011101	[0;20]	[19.4;20]	[12.2;0]	[60;0]
9860	30	0011110	[0;20]	[19.4;20]	[12.4;0]	[60;0]
9861	31	0011111	[0;20]	[19.4;20]	[12.6;0]	[60;0]
9862	32	0100000	[0;20]	[19.4;20]	[12.8;0]	[60;0]
9863	33	0100001	[0;20]	[19.4;20]	[13;0]	[60;0]
9864	34	0100010	[0;20]	[19.4;20]	[13.2;0]	[60;0]
9869	35	0100011	[0;20]	[19.4;20]	[14.2;0]	[60;0]
9870	36	0100100	[0;20]	[19.2;20]	[14.4;0]	[60;0]
9956	37	0100101	[0;20]	[19.6;20]	[11.4;0]	[60;0]
9957	38	0100110	[0;20]	[19.6;20]	[11.6;0]	[60;0]
9958	39	0100111	[0;20]	[19.6;20]	[12;0]	[60;0]
9960	40	0101000	[0;20]	[19.6;20]	[12.2;0]	[60;0]
9961	41	0101001	[0;20]	[19.6;20]	[12.4;0]	[60;0]
9962	42	0101010	[0;20]	[19.6;20]	[12.6;0]	[60;0]
9963	43	0101011	[0;20]	[19.6;20]	[12.8;0]	[60;0]
9969	44	0101100	[0;20]	[19.6;20]	[14;0]	[60;0]
10055	45	0101101	[0;20]	[19.8;20]	[11;0]	[60;0]
10056	46	0101110	[0;20]	[19.8;20]	[11.2;0]	[60;0]
10057	47	0101111	[0;20]	[19.8;20]	[11.4;0]	[60;0]
10058	48	0110000	[0;20]	[19.8;20]	[11.6;0]	[60;0]
10059	49	0110001	[0;20]	[19.8;20]	[11.8;0]	[60;0]

Таблица 4.8 – продолжение

№ _к	№ _{ГА}	№ _{CD}	B ₀	B ₁	B ₂	B ₃
10060	50	0110010	[0;20]	[19.8;20]	[12;0]	[60;0]
10061	51	0110011	[0;20]	[19.8;20]	[12.2;0]	[60;0]
10062	52	0110100	[0;20]	[19.8;20]	[12.4;0]	[60;0]
10063	53	0110101	[0;20]	[19.8;20]	[12.6;0]	[60;0]
10068	54	0110110	[0;20]	[19.8;20]	[13.6;0]	[60;0]
10069	55	0110111	[0;20]	[19.8;20]	[13.8;0]	[60;0]
10154	56	0111000	[0;20]	[20;20]	[10.6;0]	[60;0]
10155	57	0111001	[0;20]	[20;20]	[10.8;0]	[60;0]
10156	58	0111010	[0;20]	[20;20]	[11;0]	[60;0]
10157	59	0111011	[0;20]	[20;20]	[11.2;0]	[60;0]
10158	60	0111100	[0;20]	[20;20]	[11.4;0]	[60;0]
10159	61	0111101	[0;20]	[20;20]	[11.6;0]	[60;0]
10160	62	0111110	[0;20]	[20;20]	[11.8;0]	[60;0]
10161	63	0111111	[0;20]	[20;20]	[12;0]	[60;0]
10162	64	1000000	[0;20]	[20;20]	[12.2;0]	[60;0]
10163	65	1000001	[0;20]	[20;20]	[12.4;0]	[60;0]
10168	66	1000010	[0;20]	[20;20]	[13.4;0]	[60;0]

Здесь №_к – номер кривой в исходном пространстве поиска; №_{ГА} – номер кривой в конечном пространстве поиска; №_{CD} – идентификатор кривой в непрерывном кодовом пространстве (исходное кодовое пространство); B₀, B₁, B₂, B₃ – вершины определяющего кривую опорного многоугольника.

Определённое таким образом кодовое пространство решений передадим на вход генетического алгоритма. Тестирование алгоритма будем проводить на трёх независимо сформированных выборках из исходного кодового пространства, что позволит дать качественную оценку эффективности применения эволюционных методов поиска в рамках решения текущей задачи.

Настроечные параметры генетического алгоритма и информация о параметрах кодового пространства приведена в таблице 4.9, сделанные выборки показаны в таблицах 4.10, 4.11, 4.12. Пошаговая трассировка эксперимента для выборки №1 приведена в Приложении А

Таблица 4.9 – Настроечные параметры ГА

N_o	N_p	N_i	p
67	14	100	10^{-5}

Здесь N_o – количество кривых в кодовом пространстве; N_p – количество особей в поколении; N_i – количество итераций алгоритма до принудительного останова; p – точность машинных вычислений.

Таблица 4.10 – Выборка №1

№	Исходное поколение		Финальное поколение	
	CD	$F_{ц}$	CD	$F_{ц}$
0	0000001	0,01548	0111110	0,0154
1	0000100	0,01547	0011111	0,01541
2	0001001	0,01546	0011111	0,01541
3	0010001	0,01544	0011111	0,01541
4	0011010	0,01545	0111111	0,0154
5	0011101	0,01541	0011110	0,01541

Таблица 4.10 – продолжение

№	Исходное поколение		Финальное поколение	
	CD	F _ц	CD	F _ц
6	0100001	0,01543	0011110	0,01541
7	0101010	0,01542	0011111	0,01541
8	0101100	0,01544	0011111	0,01541
9	0101101	0,01539	0011110	0,01541
10	0110010	0,0154	0011111	0,01541
11	0111000	0,01538	0111110	0,0154
12	0111110	0,0154	0111111	0,0154
13	1000000	0,0154	0011110	0,01541

Таблица 4.11 – Выборка №2

№	Исходное поколение		Финальное поколение	
	CD	F _ц	CD	F _ц
0	0000011	0,01547	0000001	0,01548
1	0001100	0,01544	0000001	0,01548
2	0010011	0,01546	0000001	0,01548
3	0011111	0,01541	0000001	0,01548
4	0100110	0,0154	0000001	0,01548
5	0101011	0,01542	0000001	0,01548
6	0110001	0,0154	0000001	0,01548
7	0110010	0,0154	0000001	0,01548
8	0110100	0,01541	0000001	0,01548
9	0110111	0,01543	0000001	0,01548

Таблица 4.11 – продолжение

№	Исходное поколение		Финальное поколение	
	CD	F _ц	CD	F _ц
10	0111001	0,01538	0000001	0,01548
11	0111101	0,0154	0000001	0,01548
12	0111110	0,0154	0000001	0,01548
13	1000000	0,0154	0000001	0,01548

Таблица 4.12 – Выборка №3

№	Исходное поколение		Финальное поколение	
	CD	F _ц	CD	F _ц
0	0000001	0,01548	0000010	0,01547
1	0000010	0,01547	0000011	0,01547
2	0000100	0,01547	0000011	0,01547
3	0000101	0,01546	0000011	0,01547
4	0000110	0,01547	0000011	0,01547
5	0010010	0,01544	0000011	0,01547
6	0100101	0,0154	0000010	0,01547
7	0101011	0,01542	0000001	0,01548
8	0101101	0,01539	0000011	0,01547
9	0101110	0,01539	0000011	0,01547
10	0110010	0,0154	0000010	0,01547
11	0110111	0,01543	0000011	0,01547
12	0111000	0,01538	0000011	0,01547
13	1000010	0,01542	0000011	0,01547

Здесь № – номер особи в выборке; CD – код особи (хромосома); $F_{ц}$ – значение целевой функции.

Адаптация решений исходного и финального поколений для выборок 1-3 приведены в таблицах 4.13, 4.14, 4.15.

Таблица 4.13 – Выборка №1, адаптация

№	Исходное поколение		Финальное поколение	
	CD	$P_s, \%$	CD	$P_s, \%$
0	0001001	7,16271%	0000010	7,14286%
1	0001011	7,15345	0000010	7,14286
2	0001111	7,14418	0000010	7,14286
3	0010000	7,14881	0000010	7,14286
4	0010001	7,15345	0000010	7,14286
5	0100001	7,14881	0000010	7,14286
6	0100011	7,15345	0000010	7,14286
7	0100101	7,13491	0000010	7,14286
8	0101000	7,13491	0000010	7,14286
9	0110001	7,13491	0000010	7,14286
10	0110010	7,13491	0000010	7,14286
11	0111001	7,12565	0000010	7,14286
12	0111100	7,13491	0000010	7,14286
13	1000000	7,13491	0000010	7,14286

Таблица 4.14 – Выборка №2, адаптация

№	Исходное поколение		Финальное поколение	
	CD	P _s , %	CD	P _s , %
0	0000011	7,16801	0000001	7,14286
1	0001100	7,15411	0000001	7,14286
2	0010011	7,16338	0000001	7,14286
3	0011111	7,14021	0000001	7,14286
4	0100110	7,13558	0000001	7,14286
5	0101011	7,14484	0000001	7,14286
6	0110001	7,13558	0000001	7,14286
7	0110010	7,13558	0000001	7,14286
8	0110100	7,14021	0000001	7,14286
9	0110111	7,14948	0000001	7,14286
10	0111001	7,12631	0000001	7,14286
11	0111101	7,13558	0000001	7,14286
12	0111110	7,13558	0000001	7,14286
13	1000000	7,13558	0000001	7,14286

Таблица 4.15 – Выборка №3, адаптация

№	Исходное поколение		Финальное поколение	
	CD	P _s , %	CD	P _s , %
0	0000001	7,166	0000010	7,14253
1	0000010	7,16137	0000011	7,14253
2	0000100	7,16137	0000011	7,14253
3	0000101	7,15674	0000011	7,14253

Таблица 4.15 – продолжение

№	Исходное поколение		Финальное поколение	
	CD	P _s , %	CD	P _s , %
4	0000110	7,16137	0000011	7,14253
5	0010010	7,14749	0000011	7,14253
6	0100101	7,12897	0000010	7,14253
7	0101011	7,13823	0000001	7,14714
8	0101101	7,12434	0000011	7,14253
9	0101110	7,12434	0000011	7,14253
10	0110010	7,12897	0000010	7,14253
11	0110111	7,14286	0000011	7,14253
12	0111000	7,11971	0000011	7,14253
13	1000010	7,13823	0000011	7,14253

Здесь № – номер особи в выборке; CD – код особи (хромосома); P_s – вероятность скрещивания для каждой хромосомы текущего поколения решений (ширина сектора колеса рулетки) в процентах.

Трассировка операций кроссовера, первой итерации ГА, для выборок 1-3, представлены в таблицах 4.16, 4.17, 4.18.

Таблица 4.16 – Выборка №1, кроссовер

№	Пара родителей		Пара потомков	
	CD ₁	CD ₂	CD ₁	CD ₂
1	001 0001	010 1000	0011000	0100001
2	01 10010	01 01000	0101000	0110010

Таблица 4.16 – продолжение

№	Пара родителей		Пара потомков	
	CD ₁	CD ₂	CD ₁	CD ₂
3	0 101000	0 010000	0010000	0101000
4	0010 001	0101 000	0010000	0101001
5	0 010001	0 110001	0110001	0010001
6	01 10010	01 01000	0101000	0110010
7	0001001	0001001	0000000	1000000

Таблица 4.17 – Выборка №2, кроссовер

№	Пара родителей		Пара потомков	
	CD ₁	CD ₂	CD ₁	CD ₂
1	10 00000	10 00000	1000000	1000000
2	0111110	0111110	0111000	1000000
3	01 10001	01 01011	0101011	0110001
4	011 0001	011 0001	0110001	0110001
5	0110 001	1000 000	0110000	1000001
6	01 10010	01 01011	0101011	0110010
7	0111110	0111110	0111000	1000000

Таблица 4.18 – Выборка №3, кроссовер

№	Пара родителей		Пара потомков	
	CD ₁	CD ₂	CD ₁	CD ₂
1	010 1011	001 0010	0100010	0011011
2	011 1000	000 0100	0110100	0001000

Таблица 4.18 – продолжение

№	Пара родителей		Пара потомков	
	CD ₁	CD ₂	CD ₁	CD ₂
3	0110 010	0000 100	0110100	0000010
4	0 101110	0 110010	0110010	0101110
5	000001 0	000010 1	0000011	0000100
6	01 11000	00 00100	0100100	0011000
7	00000 10	00100 10	0000010	0010010

Здесь № – номер операции кроссовера в итерации; CD₁, CD₂ – пар хромосом родителей и потомков соответственно. Знаком «|» показана точка расщепления в родительских парах хромосом при операции скрещивания.

Процесс итерационного приближения к решению (пошагово) для выборок 1-3, приведён в таблицах 4.19, 4.20, 4.21. Каждая строка содержит информацию о лучшем решении текущего поколения.

Таблица 4.19 – Выборка №1, приближение к решению

№	F _{цс}	F _{цл}	CD
1	0,01542	0,0154	0001001
2	0,01542	0,0154	0000000
3	0,01541	0,0154	0011010
4	0,01541	0,01542	0000010
5	0,01541	0,0154	0000010
6	0,01542	0,0154	0100010
7	0,01543	0,01542	0000001

Таблица 4.19 – продолжение

№	F _{цс}	F _{цл}	CD
8	0,01542	0,01542	0000000
9	0,01542	0,0154	0000001
10	0,01544	0,01548	0000001
11	0,01544	0,01541	0000000
12	0,01545	0,01548	0000001
13	0,01545	0,01547	0000001
14	0,01546	0,01548	0000001
15	0,01546	0,01547	0000001
16	0,01546	0,01547	0000001
17	0,01547	0,01547	0000010
18	0,01547	0,01547	0000010
19	0,01547	0,01547	0000010
20	0,01547	0,01547	0000010
21	0,01547	0,01547	0000010
22	0,01547	0,01547	0000010

Таблица 4.20 – Выборка №2, приближение к решению

№	F _{цс}	F _{цл}	CD
1	0,01542	0,0154	0000011
2	0,0154	0,0154	0101011
3	0,01541	0,0154	0000000
4	0,01541	0,01543	0100001
5	0,01542	0,01543	0000001

Таблица 4.20 – продолжение

№	$F_{цс}$	$F_{цл}$	CD
6	0,01542	0,0154	0100001
7	0,01543	0,01541	0000001
8	0,01543	0,01541	0000001
9	0,01545	0,01543	0000001
10	0,01544	0,01548	0000001
11	0,01544	0,01548	0000001
12	0,01546	0,01548	0000001
13	0,01546	0,01548	0000001
14	0,01546	0,01543	0000001
15	0,01547	0,01548	0000001
16	0,01547	0,01548	0000001
17	0,01547	0,01548	0000001
18	0,01547	0,01548	0000001
19	0,01548	0,01548	0000001
20	0,01548	0,01548	0000001
21	0,01548	0,01548	0000001
22	0,01548	0,01548	0000001
23	0,01548	0,01548	0000001
24	0,01548	0,01548	0000001

Таблица 4.21 – Выборка №3, приближение к решению

№	$F_{цс}$	$F_{цл}$	CD
1	0,01543	0,01542	0000001

Таблица 4.21 – продолжение

№	$F_{цс}$	$F_{цл}$	CD
2	0,01544	0,01544	0000010
3	0,01545	0,01547	0000010
4	0,01546	0,01546	0000001
5	0,01546	0,01547	0000001
6	0,01546	0,01548	0000001
7	0,01547	0,01547	0000001
8	0,01547	0,01548	0000001
9	0,01547	0,01548	0000000
10	0,01547	0,01547	0000000
11	0,01547	0,01547	0000001
12	0,01547	0,01547	0000001

Здесь № – номер итерации; $F_{цс}$ – среднее значение функции приспособленности для поколения; $F_{цл}$ – лучшее значение целевой функции текущего поколения. CD – кодовое слово хромосомы, имеющей лучшее значение целевой функции в поколении.

Графическая иллюстрация процесса приближения к решению приведена на рисунке 4.7, на оси абсцисс показано количество циклов, потребовавшихся для выявления лучшей альтернативы из исследуемых кривых. Ось ординат отображает среднее значение целевой функции поколений решений.

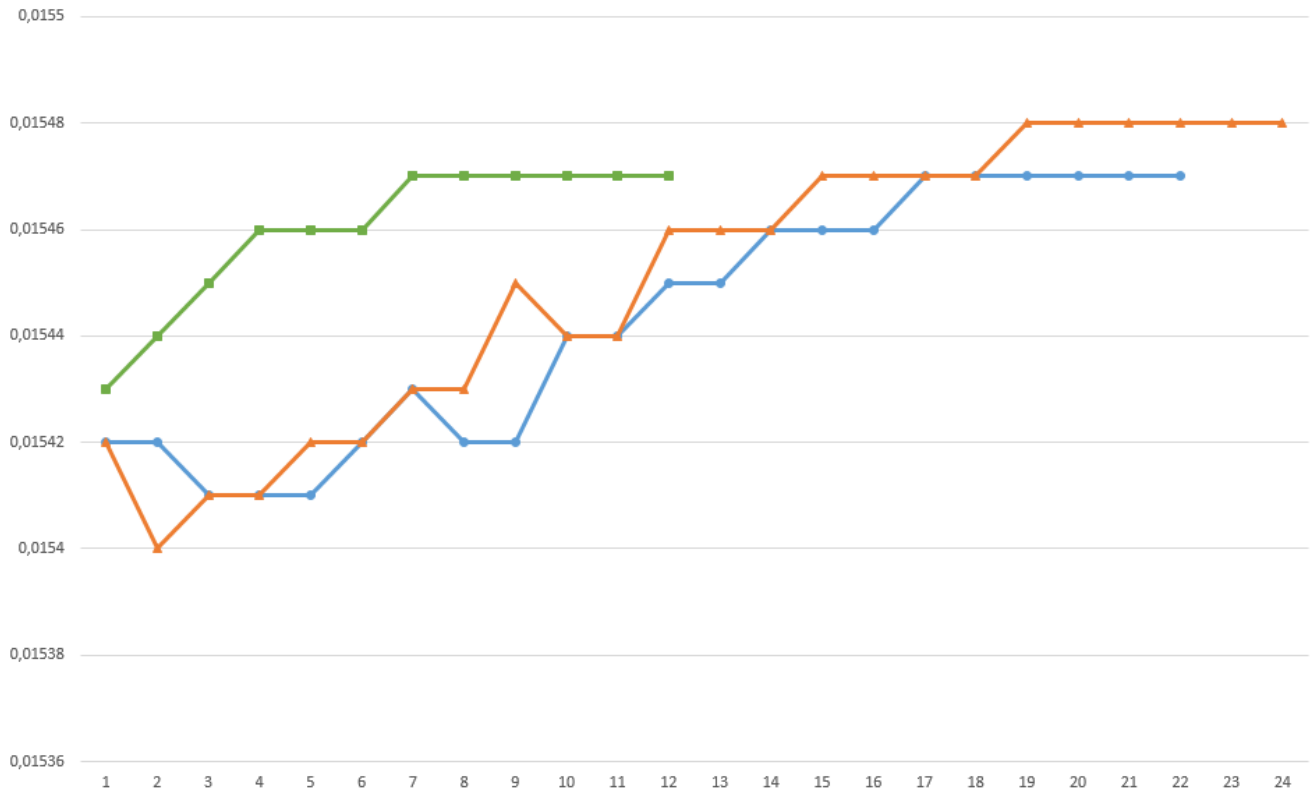


Рисунок 4.7 – Поиск кривой опорной траектории посредством ГА, приближение к решению для выборок 1-3

Результаты поиска кривой опорной траектории для всех трёх выборок сведены в таблицу 4.22. Графическое изображение найденных кривых опорных траекторий с указанием их номеров в конечном пространстве поиска приведены на рисунке 4.8.

Таблица 4.22 – Результирующие кривые

№	CD	$F_{ц}$	B_0, B_1, B_2, B_3	ITR
1	0000010	0,01547	[0;20], [18;20], [16.2;0], [60;0]	22
2	0000001	0,001548	[0;20], [17.8;20], [16.6;0], [60;0]	24
3	0000001	0,001548	[0;20], [17.8;20], [16.6;0], [60;0]	12

Здесь № – номер эксперимента; B_0, B_1, B_2, B_3 – координаты точек опорных многоугольников найденных кривых; $F_{ц}$ – значение целевой функции выбранной кривой; ITR – количество итераций генетического алгоритма.

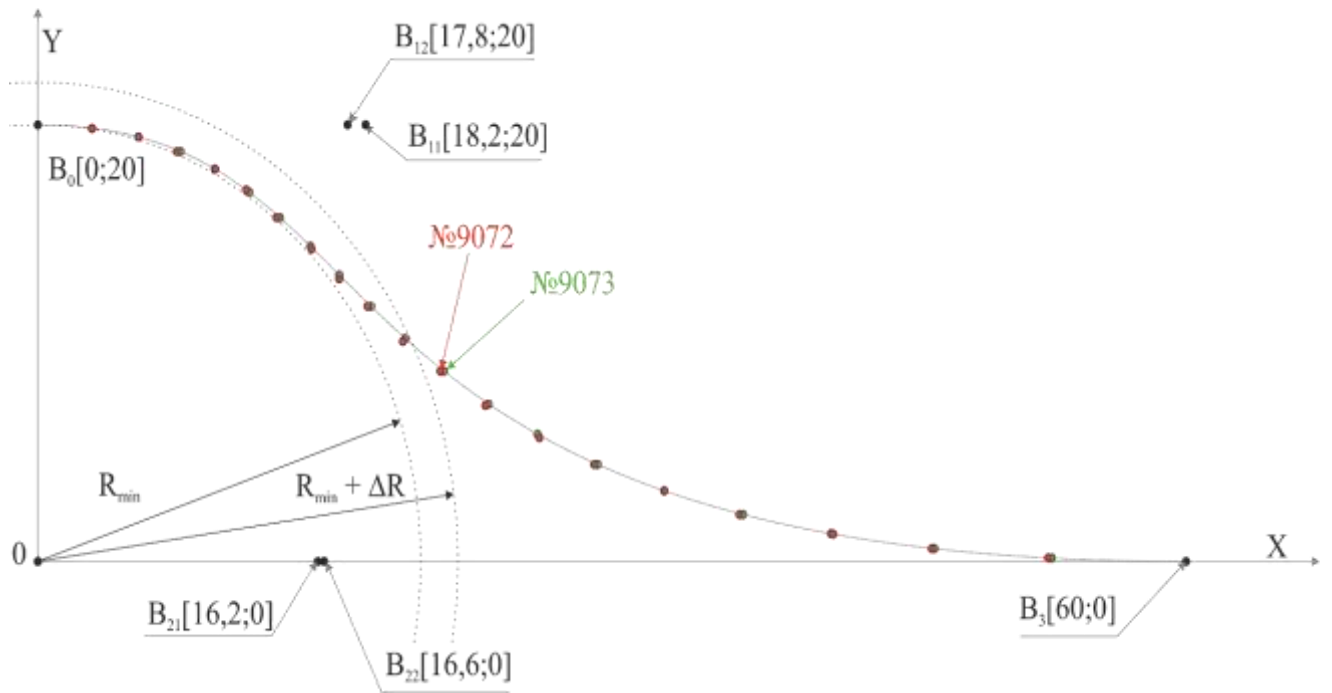


Рисунок 4.8 – Найденные кривые опорной траектории

Так как по условию вычислительного эксперимента наилучшая кривая была известна заранее, то на основании настоящего тестирования можно констатировать факт, что совокупность предложенных алгоритмов модуля ГТ справилась с поставленной задачей. Так как в двух из трёх случаев зафиксировано точное соответствие решения эталонной кривой и в одном случае в качестве решения выбрана кривая, ближайшая к эталонной (что является допустимым в силу того, что разница между кривой №9072 и №9073 пренебрежимо мала).

4.4 Полунатурные испытания, обработка траектории

Первоначальная проверка алгоритма исполнения задания осуществлялась на мобильной платформе, разработанном на базе модели автомобиля с электроприводом (КР01), шасси использованной для создания модели изображены на рисунке 4.9 (слева).

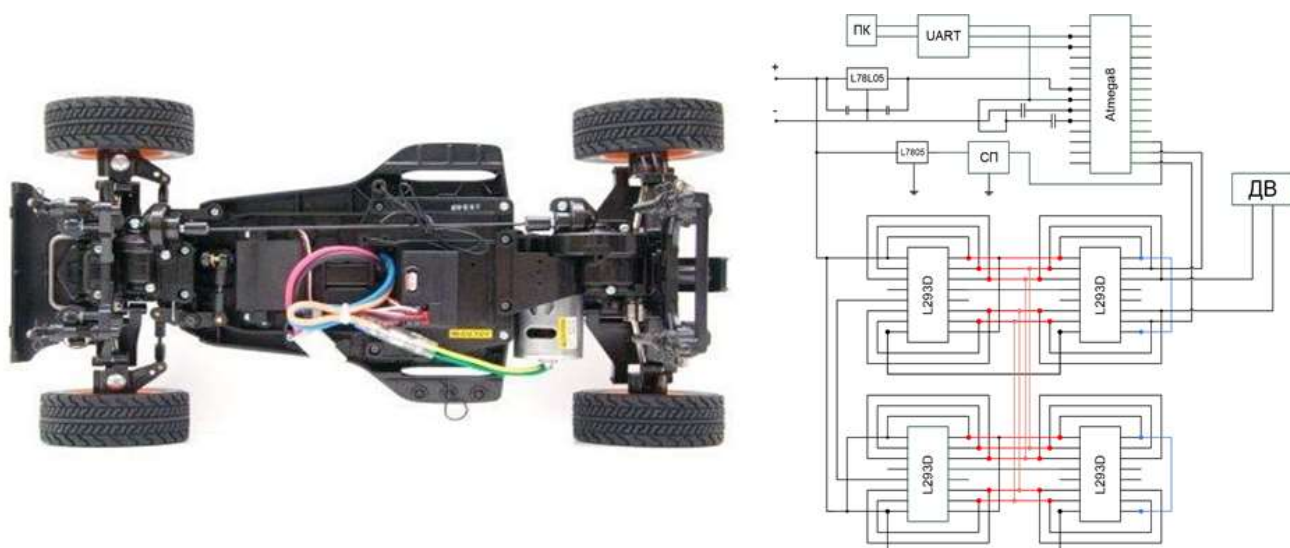


Рисунок 4.9 – Шасси модели автомобиля, использованной для реализации КР01 (слева), структурная схема системы управления (справа)

Структурная схема реализованной системы управления приведена на рисунке 4.9 (справа). Разворот управляющих колёс осуществляется сервоприводом, управляющие воздействия передаются системе управления с ПК посредством интерфейса, реализованного через UART, имитирующий COM порт. Скорость перемещения модели поддерживалась постоянной. В качестве обрабатываемого манёвра был выбран манёвр «переставка» [131]. Управляющие воздействия рассчитывались динамически при помощи специализированного программного обеспечения разработанного с помощью языка Delphi, рабочее окно разработанной программы приведено на рисунке 4.10 (слева). Разработанное приложение в качестве параметров принимает набор точек опорного многоугольника обрабатываемой кривой Безье, расчёт углов поворота колёс и времени выдачи управляющих воздействий осуществляется динамически. Данное программное обеспечение позволяет производить обработку траектории в двух режимах. В пошаговом, старт – выдача очередного управляющего воздействия – стоп, и в автоматическом, без остановок. Приложение снабжено блоком трассировки результата исполнения задания, а также блоком управления настройками задания, позволяющим определять количество выдаваемых управляющих воздействий на колёса мобильной

платформы. Во время маневрирования осуществлялась видеофиксация. Исполняемое роботом задание (опорная траектория), была предварительно рассчитана, нанесена на поверхность перемещения и размечена координатной сеткой, фото эксперимента приведено на рисунке 4.10 (справа).

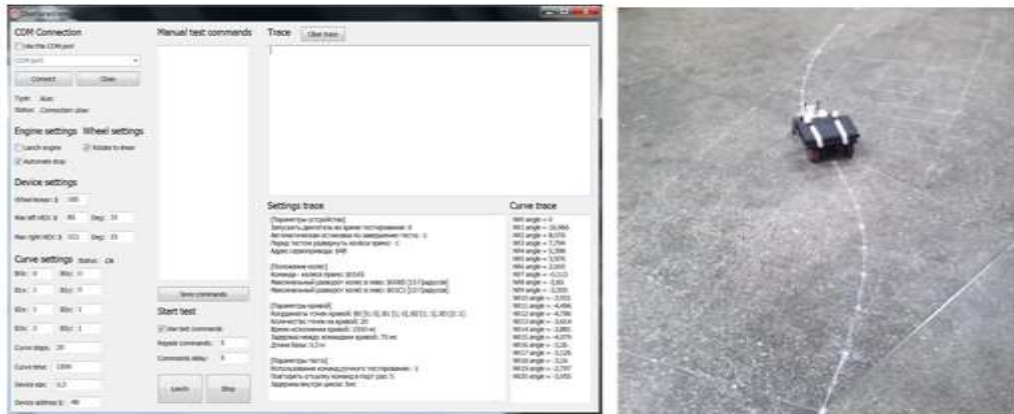


Рисунок 4.10 – Слева – интерфейс программы отработки траектории, разработанной для модели КР01; справа – фото-исполнение задания КР01, определённого опорной траекторией

Для дальнейшего исследования алгоритма отработки траектории была разработана мобильная платформа (КР02), имеющая большие габаритные размеры и массу, что позволяет дать оценку погрешностям, связанным с инерциальными свойствами реального объекта управления. Экспериментальная мобильная платформа была разработана на базе модели автомобиля с электроприводом, шасси которого показаны на рисунке 4.11.



Рисунок 4.11 – Шасси КР02

Система управления КР02 включает в себя драйвер колёс, принципиальная электрическая схема которого изображена на рисунке 4.12, и схему управления поворотом управляющих колёс, реализованной с помощью шагового двигателя.

Система драйвера колёс КР02 реализована на базе микроконтроллера atmega8535, взаимодействующего с драйверами двигателей L293D. Управление скоростью движения и углом разворота колёс КР02 осуществляется с помощью специально разработанного для решения данной задачи программного пакета для ПК.

В настоящий момент серия испытаний, запланированная для КР02, продолжается, также продолжается работа по тестированию и отладке программного обеспечения.

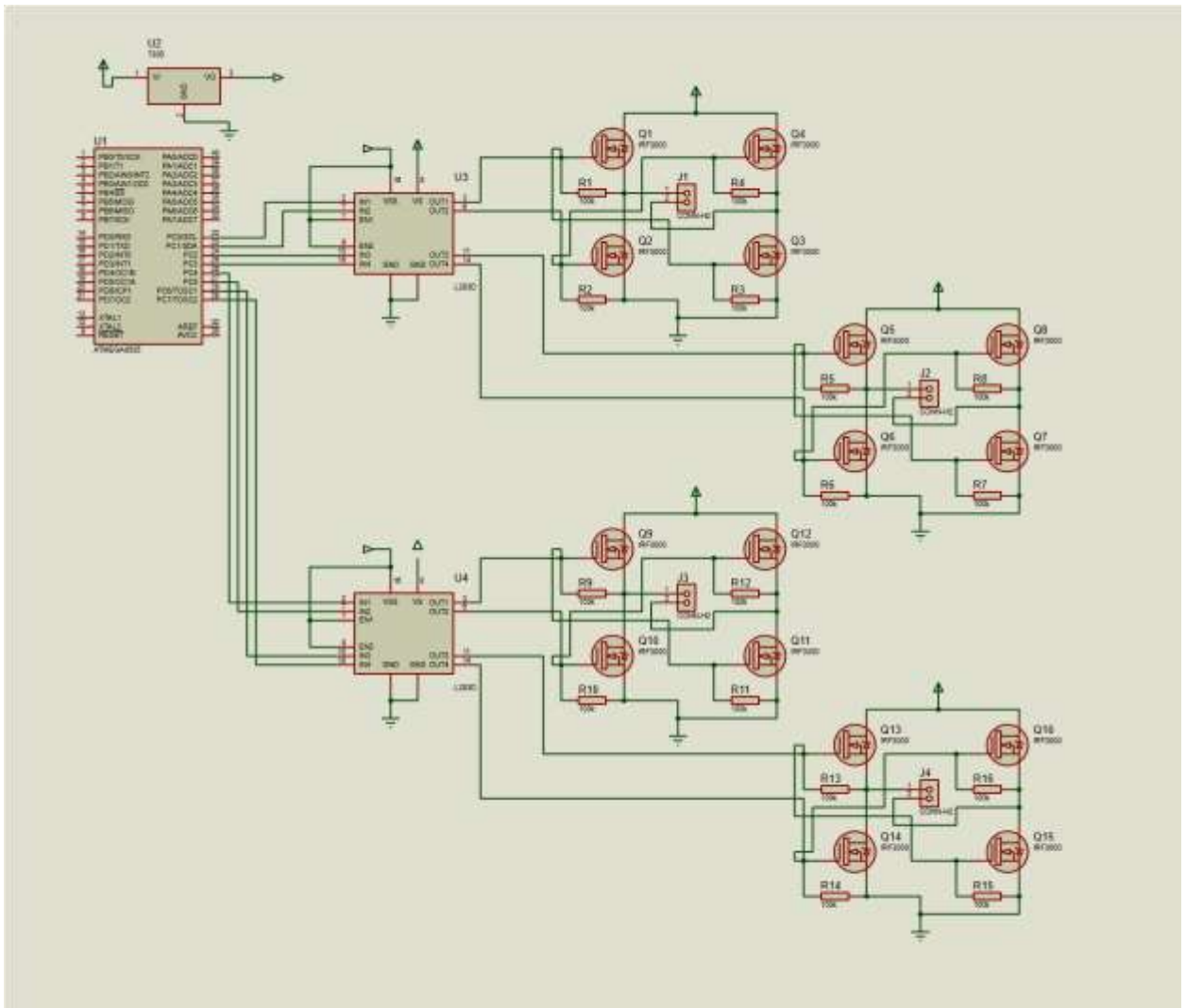


Рисунок 4.12 – Принципиальная электрическая схема драйвера колёс КР02

4.5 Экспериментальное исследование зависимости «плавности» хода мобильной платформы от количества пролётных точек в задании

При внедрении систем управления движением колёсных роботов на этапе испытаний как правило наблюдается эффект «рыскания» при отработке пролётных точек исполняемого задания в момент изменения поперечных по отношению к направлению движения ускорений, обусловленный силами реакции, действующими на робота в момент изменения направления движения. «РЫСКАНИЕ — непрерывные незначительные угловые отклонения в ту и др. сторону от основного направления движения (курса) летательного аппарата, судна, автомобиля и т.д.» [134] Рязанцев В.Д., Большая политехническая энциклопедия, с. 435. Данный эффект оказывает негативное влияние на такую эксплуатационную характеристику мобильной платформы как «плавность хода» [46], которая в свою очередь играет важную роль при транспортировке грузов по территории складского комплекса.

Существуют различные методы, направленные на снижение таких отклонений, в том числе за счёт модификацией устройства подвески транспортного средства [46]. Однако в случае если управляющие воздействия являются дискретными, то существенного снижения данного эффекта при равных прочих условиях можно добиться за счёт корректировки величины углов единовременного отклонения управляющих колёс мобильной платформы от его продольной оси. Следовательно, регулируя количество пролётных точек задания робота (количество выдаваемых управляющих на исполнительные узлы системы управления направлением движения) в течение одного цикла управления, можно добиться изменения плавности хода.

Для определения зависимости рассматриваемого типа отклонений, от количества управляющих воздействий, была проведена серия экспериментов, для двух типов экспериментальных мобильных платформ (КР01 и платформы КР02), отклонения фиксировались при помощи лазерного дальномера, снабжённого flash-

памятью и способного осуществлять запись снятых значений. Для исследования были отработаны три типа траекторий для разного количества управляющих воздействий, а именно: «дуга», «переставка» и «змейка». Результаты усреднённых значений отклонений сведены в таблицу 4.23, на основании которой для каждого случая построены графики, отражающие зависимость «плавности» хода мобильной платформы от количества пролётных точек в задании рисунки 4.13 - 4.18.

Таблица 4.23. – Результаты моделирования

Кол-во управляющих воздействий	Отклонение в м 10^{-3}					
	КР01			КРС02		
<i>N</i>	дуга	переставка	змейка	дуга	переставка	змейка
25	4,472	4,532	4,46	6,672	6,591	6,389
37	4,229	4,173	4,396	6,229	6,043	6,022
50	3,241	3,435	3,151	5,041	5,163	5,317
63	1,98	2,052	2,168	3,38	3,515	3,23
75	1,442	1,446	1,429	2,542	2,736	2,272
87	1,368	1,168	1,574	2,568	2,346	2,29
100	1,529	1,733	1,709	2,629	2,651	2,773
113	1,76	1,674	1,818	2,86	2,96	2,768
125	2,116	2,243	2,003	3,116	3,203	3,342
137	2,344	2,339	2,549	3,544	3,716	3,395
150	2,567	2,803	2,78	3,867	3,984	3,773
163	2,773	2,668	2,542	3,973	4,064	3,724
175	2,823	2,709	2,823	4,123	4,412	4,047
187	2,813	2,753	2,714	4,213	3,953	4,168
200	2,873	2,947	2,999	4,273	4,242	4,222

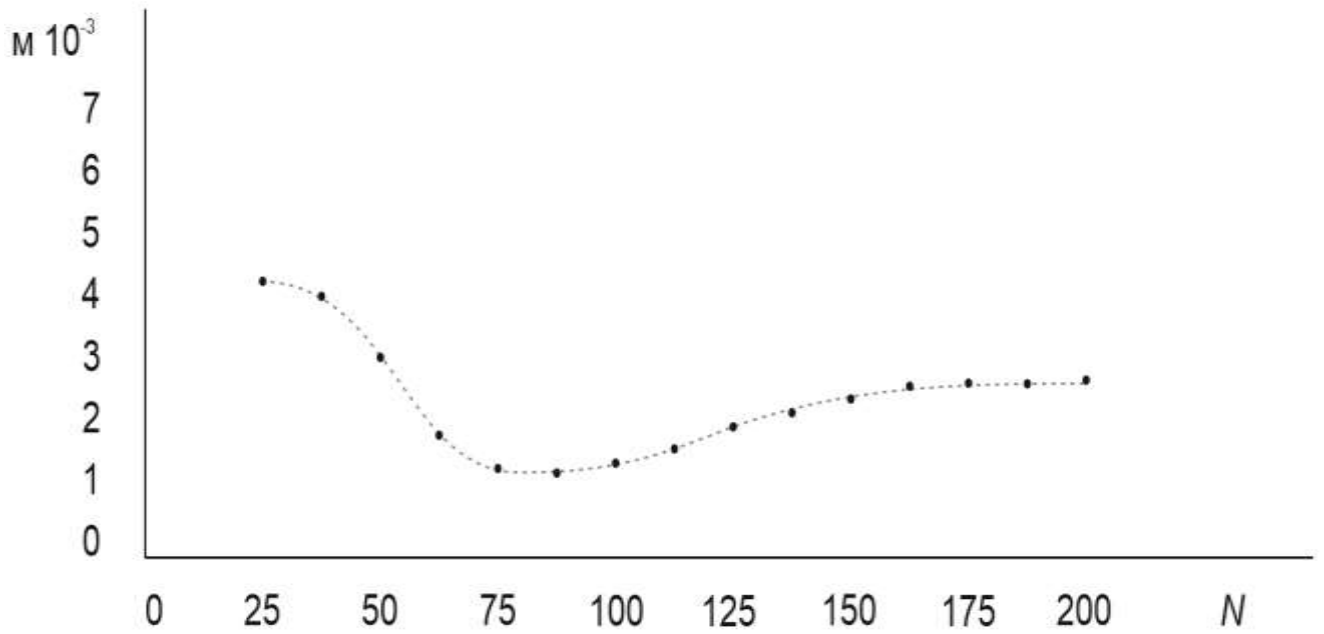


Рисунок 4.13 – График, отражающий зависимость «плавности» хода мобильной платформы от количества пролётных точек в задании. КР01, манёвр «дуга»

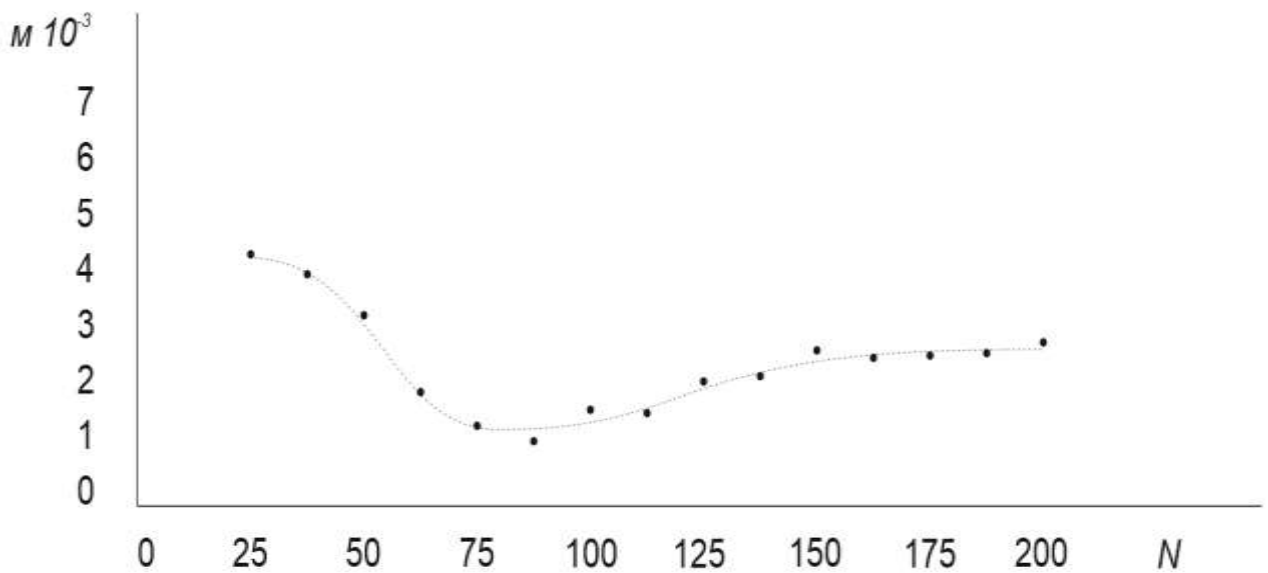


Рисунок 4.14 – График, отражающий зависимость «плавности» хода мобильной платформы от количества пролётных точек в задании. КР01, манёвр «переставка»

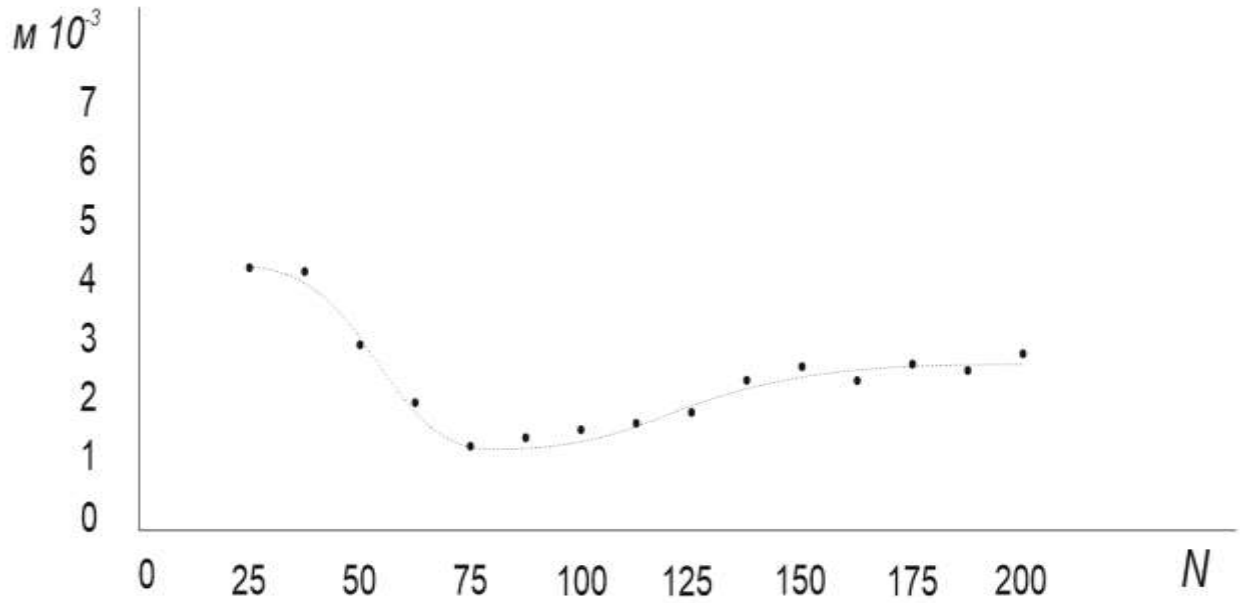


Рисунок 4.15 – График, отражающий зависимость «плавности» хода мобильной платформы от количества пролётных точек в задании. КР01, манёвр «змейка»

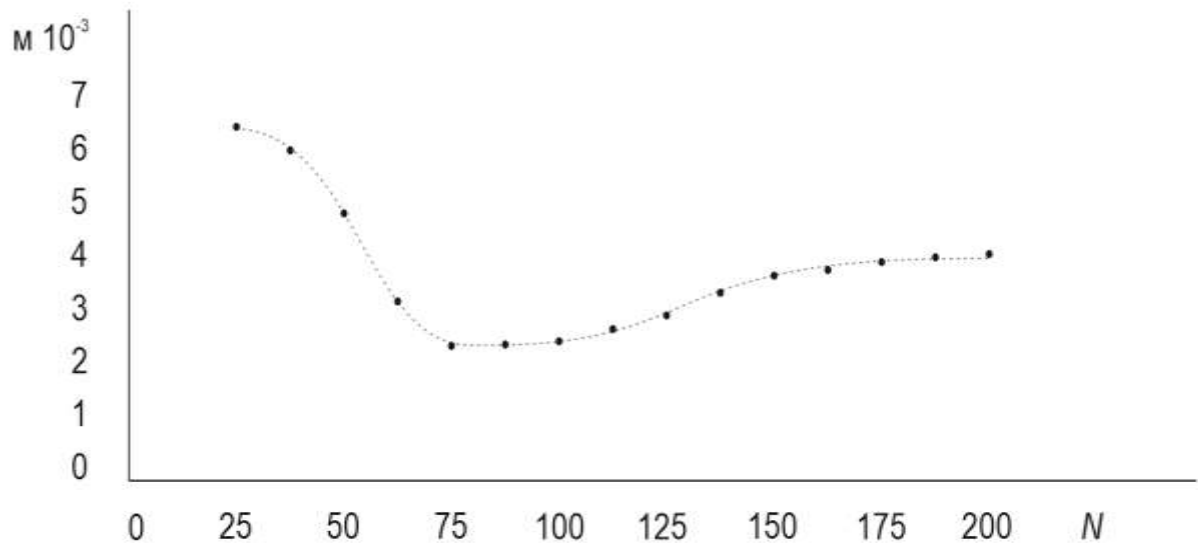


Рисунок 4.16 – График, отражающий зависимость «плавности» хода мобильной платформы от количества пролётных точек в задании. КР02, манёвр «дуга»

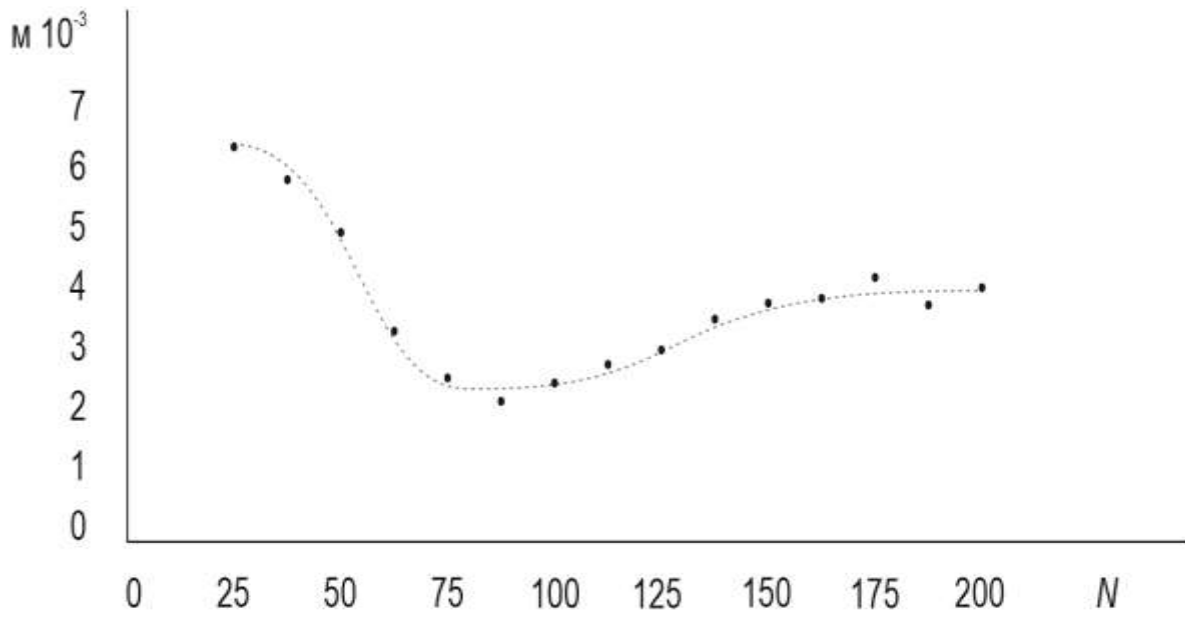


Рисунок 4.17 – График, отражающий зависимость «плавности» хода мобильной платформы от количества пролётных точек в задании. КР02, манёвр «переставка»

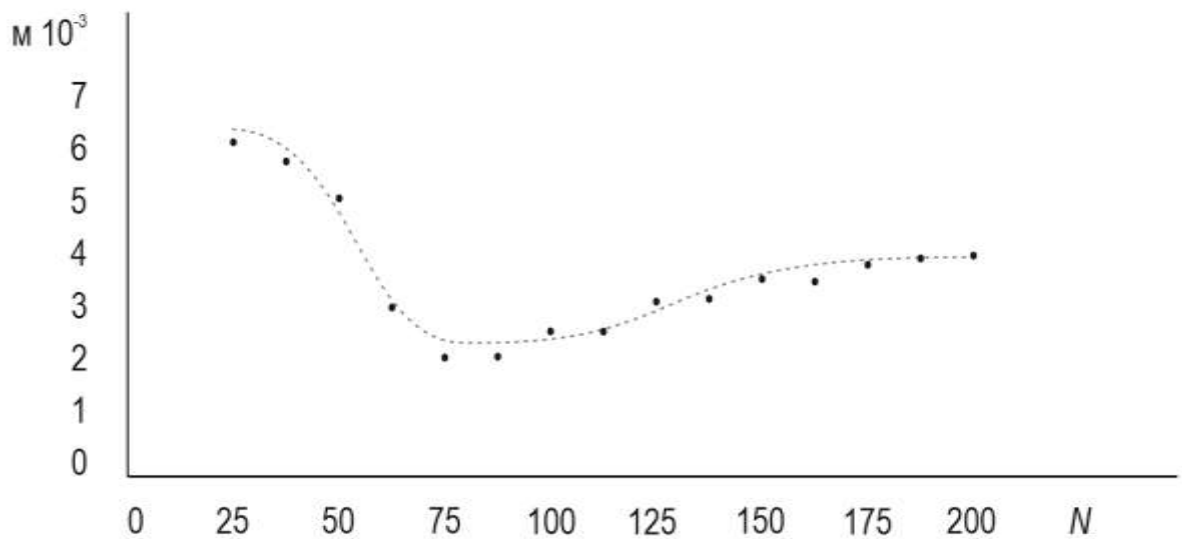


Рисунок 4.18 – График, отражающий зависимость «плавности» хода мобильной платформы от количества пролётных точек в задании. КР02, манёвр «змейка»

На основании исследования полученных данных, был сформирован график, отражающий общий характер зависимости «плавности» хода мобильной платформы от количества пролётных точек в задании, система управления движением которой реализована на основе предложенных алгоритмов (рисунок 4.19). График может быть использован на этапе внедрения системы на конкретном объекте управления с целью настройки параметров системы управления.

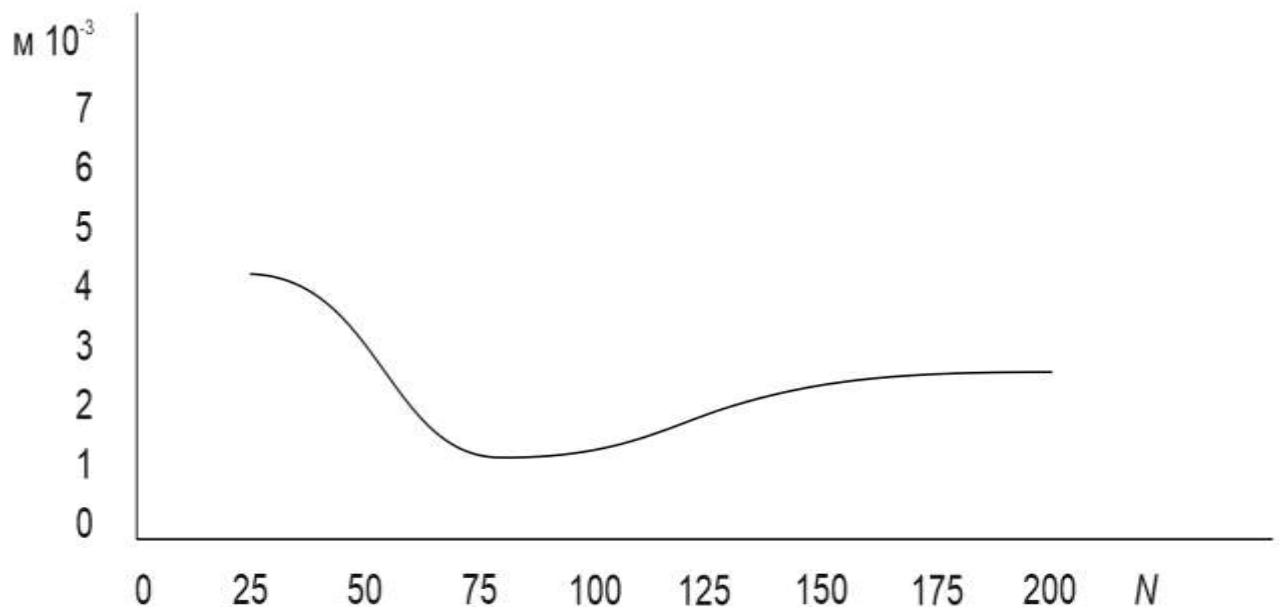


Рисунок 4.19 – Общий вид зависимости «плавности» хода мобильной платформы от количества пролётных точек в задании

Проанализировав полученный график можно сделать следующие выводы:

- При увеличении количества управляющих воздействий значения отклонений сначала резко снижаются;
- Затем наблюдается пологий участок, на котором отклонения приближаются к значениям люфтов в подвеске мобильной платформы;
- При дальнейшем увеличении количества управляющих воздействий значения отклонений начинают возрастать, что обусловлено избыточным количеством управляющих воздействий, побуждающих «дребезг» в системе.

Из графика следует вывод, что на этапе контрольных испытаний при внедрении системы управления направлением движения колёсного робота на основе предложенных алгоритмов, целесообразно установить количество управляющих воздействий, выдаваемых на исполнительные узлы системы управления в диапазоне от 75 до 100 в единицу времени.

4.6 Сравнительная оценка временных затрат, связанных с перемещением грузов

В рамках настоящего исследования была проведена сравнительная оценка временных затрат, связанных с перемещением грузов на заявленном участке транспортировки логистического центра.

Исследование проводилось с использованием комплекса имитационного моделирования RDO. В рамках формирования модели считалось, что:

- Транспортировка грузов осуществляется в рабочей зоне транспортёров, т.е. от участка первоначальной приёмки грузов до места сортировки грузов;
- В перемещении грузов задействованы 30 транспортёров. В первом случае транспортёры находятся под управлением человека-оператора. Во втором случае перемещение грузов осуществлялось таким же количеством роботизированных ричтраков;
- Условия, в которых осуществлялась транспортировка, в обоих случаях были одинаковыми.

Имитация человека-оператора была реализована путём введения дополнительных ошибок-задержек, распределение во времени которых имело случайный характер. Типы введённых ошибок-задержек:

- неверно выбранный маршрут (не кратчайший),

- простой (время от получения до начала выполнения задания),
- ошибка доставки груза (доставка к неверному терминалу).

В рамках имитации для каждого случая было выполнено по 300 заданий. Результаты, полученные посредством моделирования, сведены в таблицу 4.24, а их графическая интерпретация проиллюстрирована на рисунке 4.20.

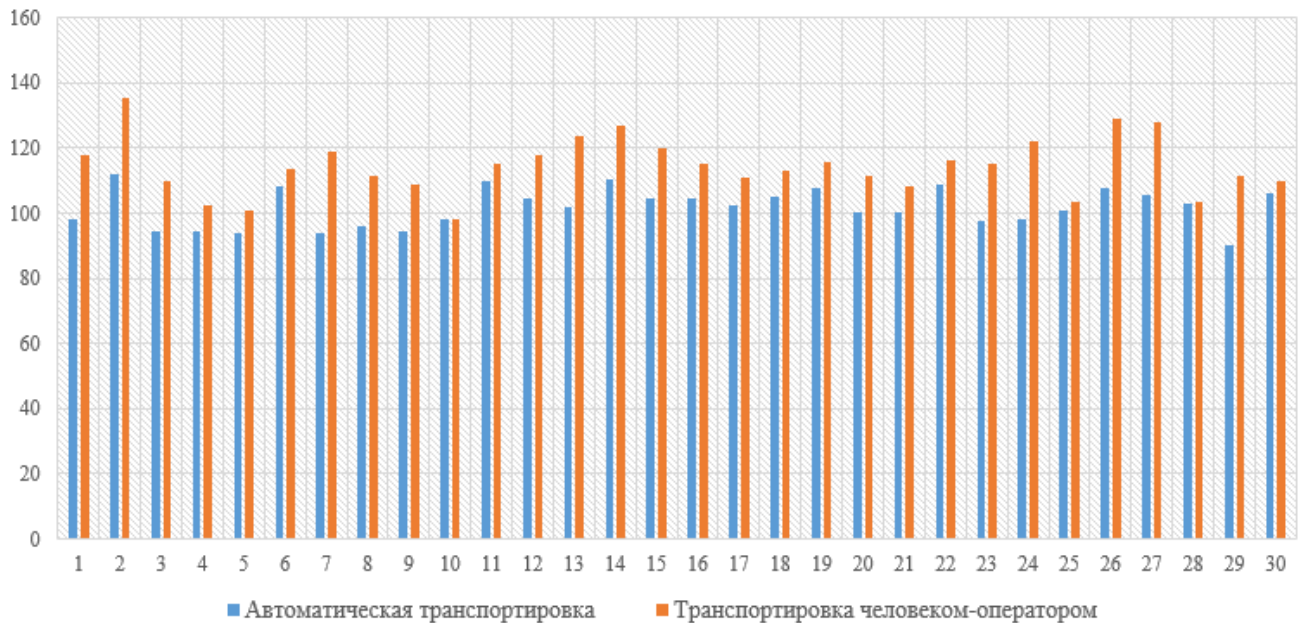


Рисунок 4.20 – Сравнение затрат времени, затраченного на транспортировку объектов складирования

Таблица 4.24 – Результаты имитационного моделирования

№	T ₁	T ₂	№	T ₁	T ₂	№	T ₁	T ₂
1	97,94	117,64	11	110,03	115,21	21	100,34	108,04
2	111,71	135,45	12	104,37	117,89	22	108,8	116,04
3	94,35	109,89	13	101,79	123,51	23	97,67	115,23
4	94,26	102,3	14	110,52	126,87	24	98,3	121,92
5	94,11	100,73	15	104,72	119,71	25	101,04	103,34
6	108,47	113,58	16	104,73	115,14	26	107,57	128,75
7	93,67	119	17	102,47	110,99	27	105,41	127,98
8	95,75	111,21	18	105,2	113,03	28	103,14	103,71
9	94,52	108,77	19	107,61	115,63	29	90,22	111,48
10	98,3	98,29	20	100,47	111,7	30	105,93	110,07

Здесь № – номер эксперимента, T1 – суммарное время в минутах, затраченное на выполнение всех транспортных задач группой роботов, T2 – суммарное время в минутах, затраченное на выполнение всех транспортных задач группой транспортёров под управлением человека-оператора.

По результатам проведённого имитационного моделирования определено, что при среднем времени выполнения набора заданий 114.44 минуты, группа роботизированных транспортёров справляется с заданием на 12.66 минуты быстрее. Таким образом, автоматизация перемещения объектов хранения обеспечивает сокращение времени обработки грузов на 11.06%, что в условиях реального логистического центра является существенным.

4.7 Выводы по главе 4

- Проведена серия вычислительных экспериментов, целью которых являлось исследование предложенного алгоритма исполнения задания на предмет его пригодности к применению для реализации системы управления колёсным роботом в условиях автоматизированного складского комплекса. По результатам проведённых контрольных испытаний сделан вывод о том, что разработанный алгоритм является пригодным для отработки сложных криволинейных траекторий при малых скоростях движения мобильной платформы.

- Проведено экспериментальное исследование алгоритмического обеспечения модуля генератора опорных траекторий. По результатам вычислительного эксперимента установлено, что разработанная группа алгоритмов позволяет успешно решать задачу поиска кривой опорной траектории для выполнения манёвров уклонения от столкновения с динамическим препятствием с учётом принятых в работе допущений и ограничений.

- Разработан опытный образец мобильной платформы на базе модели автомобиля с электроприводом, а также программное обеспечение, реализующее

предложенный алгоритм исполнения задания. Проведена серия натурных испытаний, во время которых модель мобильной платформы исполняла задания, определённые предварительно рассчитанными эталонными траекториями в автоматическом режиме. Основываясь на результатах наблюдений процесса выполнения манёвров, сделан вывод о том, что предложенный алгоритм исполнения задания пригоден для решения задач, связанных с управлением направлением движения колёсных роботов.

Заключение

В рамках настоящей диссертационной работы получены следующие основные результаты:

1. Выполнена научно-квалификационная работа, в которой содержится решение научной задачи, позволяющее сократить временные затраты на транспортировку объектов хранения по территории логистического центра на 11.06% за счёт применения мобильных колёсных роботов-транспортёров, реализующих планирование и исполнение заданий посредством алгоритмов основанных на методе опорных траекторий и кривых Безье третьего порядка.

2. Установлена связь между траекторией движения мобильного колёсного робота и кривыми Безье третьего порядка, обусловленная, жёсткими механическими связями, характерными для подвесок мобильных платформ реализующих четырёхколёсную кинематическую схему с автомобильной компоновкой колёс.

3. На основании установленной связи разработан алгоритм планирования маршрута мобильного робота, особенностью которого является использование кривых Безье третьего порядка в качестве опорной кривой для построения маршрута робота.

4. Разработан алгоритм выбора маршрута выполнения задания мобильным роботом из множества альтернативных маршрутов, определённых кривыми Безье третьего порядка. Особенность предложенного алгоритма заключается в использовании эволюционных методов поиска на заданном множестве решений.

5. Разработан алгоритм определения пролётных точек маршрута мобильного робота, заданного кривой Безье третьего порядка.

6. Разработан алгоритм автоматического расчёта набора управляющих воздействий, необходимых для отработки мобильным роботом маршрута, на основании массива пролётных точек, формирующих маршрут.

7. Эмпирически определен график зависимости изменения «поперечной плавности хода» во время исполнения задания мобильной платформой от количества управляющих воздействий, выданных на исполнительные узлы системы управления направлением движением робота, на основании которого выработаны инженерные рекомендации по определению количества управляющих воздействий при внедрении систем управления траекторией движением мобильной платформы на базе разработанных алгоритмов.

4. Проведена оценка влияния динамических параметров мобильной платформы на продольное отклонение от заданной траектории в рамках определённого диапазона скоростей (1,3м/с – 2,78м/с). В результате проведённого моделирования установлено, что при перемещении мобильной платформы на 20м отклонение, возникающее вследствие влияния динамических характеристик, находится в диапазоне $8 \cdot 10^{-3} \text{ м} - 2,1 \cdot 10^{-3} \text{ м}$, что не превышает допустимых значений, определённых требованиями, налагаемыми перемещаемыми грузами.

5. Произведена оценка изменения временных затрат на транспортировку объектов хранения по территории складского комплекса при внедрении предложенной системы автоматизации управления движением транспортёров. Результаты моделирования показали, что автоматизация перемещения грузов приведёт снижению временных затрат на 11.06%.

6. Создан программный пакет, реализующий разработанные алгоритмы, а также позволяющий осуществлять отладку и настройку аппаратной части на этапе создания прототипов мобильных платформ, в том числе осуществлять трассировку пролётных точек маршрута.

Список сокращений и условных обозначений

AGV	автономный транспортный модуль
GLV	Global Live View
PAN-Robots	Plug&Navigate robots for smart factories
АСТ	автоматическая самоходная тележка
ГА	генетический алгоритм
ГАП	гибкие автоматизированные производства
ГТ	модуль генератор траектории
СМ	сигнальный модуль
СУВ	система управления верхнего уровня
ФИПП	модуль формирования исходного пространства поиска
ФКПП	модуль формирования конечного пространства поиска
ФУС	модуль формирования управляющих сигналов
ЦВС	Центральная вычислительная система

Список терминов

1. базовый радиус: Расстояние между точками касания колёс мобильной платформы, расположенных на одной продольной оси поверхности перемещения.
2. ведущие колёса: Колеса, способствующие преобразованию крутящего момента, передаваемого на них через трансмиссию и полуоси, в силу тяги, то есть, в силу, вызывающую поступательное движение мобильной платформы.
3. исходное пространство поиска: Совокупность всех кривых, каждая из которых определяет маршрут робота, потенциально пригодный для выполнения текущего задания, с учётом принятых в работе предположений и ограничений, но не учитывающий возможные коллизии с препятствиями.
4. итерация: Одноразовое исполнение последовательности действий, определённых программой.
5. колесо рулетки: Эволюционный метод поиска, применяемый в составе генетических алгоритмов для селекции хромосом.
6. колёсный модуль: Сборочная единица, включающая колесо и дополнительные конструктивные элементы, с помощью которых колесо крепится к мобильной платформе.
7. конечное пространство поиска: Исходное пространство поиска, из которого исключены все кривые, определяющие маршруты, которые потенциально могут привести к столкновению с препятствием.
8. маршрут: Упорядоченное множество пространственных расположений. ГОСТ Р 60.0.0.4–2018.
9. минимальный радиус разворота. Минимальный радиус поворота транспортного средства, при движении по которому не наблюдается движения юзом.
10. мобильная платформа: Совокупность всех компонентов мобильного робота, обеспечивающих его передвижение. ГОСТ Р 60.0.0.4–2018.

11. наилучшая опорная кривая: Опорная траектория, позволяющая выполнить задание за минимальное время.
12. опорная траектория: Кривая, задающая маршрут робота.
13. пространственное расположение: Комбинация позиции и ориентации в пространстве. ГОСТ Р 60.0.0.4–2018.
14. ричтрак: Транспортёр-погрузчик, используемый для перемещения грузов по территории складских предприятий.
15. система управления верхнего уровня: Система управления, координирующая работу всех мобильных роботов (а также других автоматизированных систем) на территории складского комплекса.
16. фаза активного уклонения: Участок маршрута, на котором возможно столкновение с препятствием.
17. фаза релаксации: Участок маршрута, на котором препятствия отсутствуют.
18. Популяция – это конечное множество особей [125].
19. Особи, входящие в популяцию, в генетических алгоритмах представляются хромосомами с закодированным в них множествами параметров задачи, т.е. решений, которые иначе называются точками в пространстве поиска (search points). В некоторых работах особи называются организмами [125].
20. Хромосомы (другие названия – цепочки, или кодовые последовательности) – это упорядоченные последовательности генов [125].
21. Ген (также называемый свойством, знаком или детектором) – это атомарный элемент генотипа, в частности, хромосомы [125].
22. Генотип или структура – это набор хромосом данной особи. Следовательно, особями популяции могут быть генотипы либо единичные хромосомы (в довольно распространенном случае, когда генотип состоит из одной хромосомы) [125].

Список литературы

1. Тимофеев, А.В. Адаптивные робототехнические комплексы / А.В. Тимофеев — Л.: Машиностроение. Ленингр. отд-ние, 1988. — 332 с.
2. 2. History of Industrial Robots: From the first installation until today. [Электронный ресурс] // IFR, International Federation of Robotics. – 2019. Режим доступа: https://web.archive.org/web/20121224213437/http://www.ifr.org/uploads/media/History_of_Industrial_Robots_online_brochure_by_IFR_2012.pdf
3. Smith, R. South Korea has the highest density of robot workers in the world [Электронный ресурс] / R. Smith // World Economic Forum. — 2018. Режим доступа: <https://www.weforum.org/agenda/2018/04/countries-with-most-robot-workers-per-human/>
4. Хлебинских, Л.В. Автоматизация производства в современном мире / Л.В. Хлебинских, М.А. Зубкова, Т.Ю. Сайкова // Молодой учёный. — 2017. — №16 С. 308-310.
5. Алесинская, Т.В. Основы логистики. Функциональные области логистического управления [Электронный ресурс] / Т.В. Алесинская // Таганрог: Изд-во ТТИ ЮФУ. — 2009. — 79 с. Режим доступа: <http://www.aup.ru/books/m192/>
6. Палагута, К.А. Автоматизация процесса транспортировки объектов хранения по территории логистического центра. / К.А. Палагута, А.А. Алексеев // Сборник трудов IX Международной научно-практической конференции «Информационные и коммуникационные технологии в образовании, науке и производстве» – Протвино, АО «НПО «Турботехника». – 2016. – 507с. Сс. 298–301.
7. Классификация складских помещений [Электронный ресурс] // KnightFrank — 2018. — Режим доступа: <http://www.knightfrank.ru/resources/pdf/research/ind.pdf>
8. Автоматизированные склады для паллет [Электронный ресурс] // Группа компаний БелСКС, официальный сайт. — 2018. – Режим доступа:

<https://belsks.by/automated-systems/141-avtomatizirovannye-palletnye-sklady> (дата доступа 07.12.2016)

9. Тюльменков, В.Н. Автоматизация и оптимизация функционирования складского хозяйства распределительного центра: диссертация канд. техн. наук: 18.06.2009 / Тюльменков Владимир Николаевич — Томск: Томский государственный университет систем управления и радиоэлектроники, — 2009. — 131 с.

10. Автоматизация и роботизация складов. Складские роботы [Электронный ресурс] // Инжиниринговая компания «Меридиан». — 2018. Режим доступа: <https://m-robots.ru/info/avtomatizaciya-skladov-i-roboty/>

11. Домброва, Е. Роботы начали промышленную революцию в Петербурге [Электронный ресурс] / Е. Домброва // РБК Официальный сайт — 2018. Режим доступа: https://www.rbc.ru/spb_sz/08/01/2018/5a5326ac9a794786d1d9d26f

12. Протасов, С. Вилочный погрузчик Алгоритм выбора [Электронный ресурс] / С. Протасов // Журнал «Склад и Техника» – 2006. – №3 Режим доступа: http://www.sitmag.ru/article/azbuka/2006_03_A_2006_05_11-13_48_49/ [дата доступа 07.12.2016]

13. Маисеев, В. Когда искусственный интеллект победит магнитную разметку логистических комплексов [Электронный ресурс] / В. Маисеев // Завтра облачно: Журнал об IT-бизнесе, технологиях и цифровой трансформации — 2019. Режим доступа: <https://mcs.mail.ru/blog/kogda-iskusstvennyj-intellekt-pobedit-magnitnuyu-razmetku-logisticheskikh-kompleksov>

14. Автоматизированные склады – будущее уже наступило [Электронный ресурс] // Аналитическое издание CRE.ru — 2018. Режим доступа: <https://www.cre.ru/analytics/72027>

15. Amazon Robotics to open Massachusetts innovation hub in 2021 [Электронный ресурс] // The RobotReport.com — 2019. Режим доступа: <https://www.therobotreport.com/amazon-robotics-open-massachusetts-innovation-hub/>

16. Wurman, R.P. Coordinating Hundreds of Cooperative, Autonomous Vehicles in Warehouses / R.P. Wurman, R. D'Andrea, M. Mountz. // AI Magazine – 2008. – Vol.29, no. 1. – Pp. 9–20.

17. Wulfraat, M. Is Kiva Systems a Good Fit for Your Distribution Center? An Unbiased Distribution Consultant Evaluation. [Электронный ресурс] / M. Wulfraat // MWPVL. — 2012. Режим доступа: http://www.mwpvl.com/html/kiva_systems.html
18. Роботы «Ронави» появятся на российских складах уже в 2019 году [Электронный ресурс] // АО «РОСНАНО» — 2019. Режим доступа: <https://www.rusnano.com/about/press-centre/news/20190416-fiop-roboty-ronavi-poyavyatsya-na-rossiyskikh-skladakh-uzhe-v-2019-godu>
19. Иван Бородин (Ronavi Robotics): Проблемы, стоящие перед складами, можно решить только с помощью глобальной роботизации и автоматизации [Электронный ресурс] // Интернет-издание Robogeek.com — 2019. Режим доступа: <http://www.robogeek.ru/intervyu/problemy-stoyaschie-pered-skladami-mozhno-reshit-tolko-s-pomoschyu-robotizatsii-i-avtomatizatsii>
20. Ронави Роботикс [Электронный ресурс] // Официальный сайт. — 2019. Режим доступа: <http://ronavi-robotics.ru/h1500>
21. Vincent, J. Welcome to the automated warehouse of the future – How British supermarket Ocado is using robots to make online grocery shopping faster [Электронный ресурс] / J. Vincent // Vox Media. — 2018. Режим доступа: <https://www.theverge.com/2018/5/8/17331250/automated-warehouses-jobs-ocado-andover-amazon>
22. Elfes, A. Using occupancy grids for mobile robot perception and navigation / A. Elfes // Computer. – 1989. – Vol. 22. no.6.– Pp. 46-57.
23. Albrecht, C. Kroger Looks to Build Next Robotic Warehouse in Dallas, Texas [Электронный ресурс] / C. Albrecht // THE SPOON. — 2019. Режим доступа: <https://thespoon.tech/kroger-looks-to-build-next-robotic-warehouse-in-dallas-texas/>
24. Sabbattini, L. Advanced Sensing and Control Techniques for Multi AGV Systems in Shared Industrial Environments [Электронный ресурс] / L. Sabbattini, ed. // IEEE ETFA, Luxembourg. – 2015. Режим доступа: <http://www.pan-robots.eu/deliverables/publications/#sthash.0UBGgFkC.dpuf>

25. Boehning, M. Improving safety and efficiency of AGVs at warehouse black spots / M. Boehning // Intelligent Computer Communication and Processing (ICCP), IEEE Inter-national Conference - Cluj Napoca. – 2014. – Pp. 245 – 249
26. Мартыненко, Ю.Г. Управление движением мобильных колёсных роботов. / Ю.Г. Мартыненко // —М.: Издательский дом «Открытые системы». – 2005. – С.30-80.
27. Алексеев, А.А. Управление траекторией движения, автоматизированного внутрицехового транспортного средства / А.А. Алексеев // Естественные и технические науки. — 2014. — №12. — С. 298-300. (Из списка ВАК)
28. Cardarelli, E. Multisensor Data Fusion for Obstacle Detection in Automated Factory Logistics [Электронный ресурс] / E. Cardarelli, L. Sabattini, C. Secchi, C. Fantuzzi // Официальный сайт проекта PAN-Robots. — 2014. Режим доступа: <http://www.pan-robots.eu/deliverables/publications/#sthash.wshPngYj.dpuf>
29. Sabattini, L. Technological Roadmap to Boost the Introduction of AGVs in Industrial Applications. [Электронный ресурс] / L. Sabattini, V. Digani, C. Secchi, G. Cotena // Department of Engineering Sciences and Methods (DISMI) University of Modena and Reggio Emilia via G. Amendola. — 2014. — Pp. 203-208. Режим доступа: <http://www.pan-robots.eu/wp-content/uploads/2013/11/06646109.pdf>
30. Aikio, M. Omnidirectional vision system for robot navigation [Электронный ресурс] / M. Aikio // Finnish Optics Days. — 2013. Режим доступа: http://www.pan-robots.eu/wp-content/uploads/2013/05/Aikio_abstract.pdf
31. Aikio, M. Fisheye optics for omnidirectional perception / M. Aikio, M. Kumpulainen // Intelligent Computer Communication and Processing (ICCP), 2014 IEEE International Conference - Cluj Napoca. — 2014. — Pp. 259-263
32. Beinschob, P. Strategies for Contour-Based Self-Localization in Large-Scale Modern Warehouses [Электронный ресурс] / P. Beinschob, C. Reinke, // ICCP 2013. — 2013. Режим доступа: <http://www.pan-robots.eu/deliverables/publications/#sthash.0UBGgFkC.dpuf>

33. Aikio, M. Omnidirectional camera [Электронный ресурс] / M. Aikio, J.T. Mäkinen, B. Yang // ICCP 2013. — 2013. — Pp. 217-221. Режим доступа: <http://www.pan-robots.eu/wp-content/uploads/2013/11/06646111.pdf>
34. Nagy A.E., Development of an Om-nidirectional Stereo Vision System [Электронный ресурс] / A.E. Nagy, I. Szakats, T. Marita, S. Nedevschi // ICCP 2013. — 2013. — Pp. 235-242. Режим доступа: <http://www.pan-robots.eu/wp-content/uploads/2013/11/06646114.pdf>
35. Stimming, C. Multi-level On-board Data Fusion for 2D Safety Enhanced by 3D Perception for AGVs [Электронный ресурс] / C. Stimming, ect. // ICCP 2015. — 2015. — Pp. 239-244. Режим доступа: <http://www.pan-robots.eu/wp-content/uploads/2013/02/multi-level.pdf>
36. Sabattini, L. Advanced Sensing and Control Techniques for Multi AGV Systems in Shared Industrial Environments [Электронный ресурс] / L. Sbattini, ect. // University of Modena and Reggio Emilia. — 2015. — Pp. 1-7 Режим доступа: http://www.pan-robots.eu/wp-content/uploads/2015/09/panrobots_etfa.pdf
37. Koditschek, D.E. Task encoding: toward a scientific paradigm for robot planning and control / D.E. Koditschek // Robotics and Automation systems. — 1992. — Vol.9. — Pp. 1-2.
38. Sabattini, L. An automatic approach for the generation of the roadmap for multi-agv systems in an industrial environment. [Электронный ресурс] / L. Sabattini, ect. // IROS 2014 Chicago. — 2014.— Pp. 1-6. Режим доступа: URL: <http://www.pan-robots.eu/wp-content/uploads/2015/05/IROS2014.pdf>
39. Палагута, К.А. Автоматизация процесса транспорти-ровки объектов хранения по территории логистического центра. / К.А. Палагута, А.А. Алексеев // Сборник трудов IX Международной научно-практической конференции «Информаци-онные и коммуникационные технологии в образовании, науке и производстве» – Протвино, АО «НПО «Турботехника». – 2016. – 507с. С. 298-301.
40. Digani, V. Ensemble Co-ordination Approach in Multi-AGV Systems Applied to Industrial Warehouses [Электронный ресурс] / V. Digani, L. Sabattini, C. Car-

darelli, C. Secchi, C. Fantuzzi // Automation Science and Engineering, IEEE Transactions. – 2015. – Vol.12, no.3. – Pp. 922-934. Режим доступа: <http://www.pan-robots.eu/deliverables/publications/#sthash.0UBGgFkC.dpuf>

41. Automating logistics for the factory of the future [Электронный ресурс] // phys.org – 2019. Режим доступа: <https://phys.org/news/2015-04-automating-logistics-factory-future.html>

42. Laser Guide forklift Agv China Wholesale, China Suppliers - IKV [Электронный ресурс] // IKVROBOT – 2019. Режим доступа: <https://www.ikvrobot.com/product/agv/laser-guidance-forklift-AGV.html>

43. RoboCV [Электронный ресурс] // Официальный сайт производителя – 2019. Режим доступа: <http://robocv.ru/>

44. Шустиков, В. Робот-погрузчик от RoboCV: автоматизация до 70% всех складских процессов / В. Шустиков // Сколково официальный портал – 2019. Режим доступа: https://sk.ru/news/b/press/archive/2017/07/06/robotpogruzchik-ot-robocv-avtomatizaciya-do-70_2500_-vseh-skladskih-processov.aspx

45. ГОСТ Р 60.0.0.4-2019 / ИСО 8373:2012 Роботы и робототехнические устройства. Термины и определения [Электронный ресурс] // Режим доступа: <http://docs.cntd.ru/document/1200162703>

46. Тарасик, В.П. Теория движения автомобиля: Учебник для вузов. / В.П. Тарасик - СПб.: БХВ-Петербург, 2006. – 478с.

47. Алексеев, А.А. Автоматический объезд препятствий внутрицеховым транспортом / А.А. Алексеев и др. // Естественные и технические науки. – 2014. – №12 С.301-306. (Из списка ВАК)

48. ПОЛОЖЕНИЕ - ОХРАНА ТРУДА ПРИ СКЛАДИРОВАНИИ МАТЕРИАЛОВ ПОТ РО-14000-007- (утв- Минэкономки РФ от 25-02-98) [Электронный ресурс] / – 2019. Режим доступа: <http://www.zakonprost.ru/content/base/part/72505>

49. Focus S 350 Datasheet [Электронный ресурс] / FARO Technologies, Inc. – 2019. – 2с. Режим доступа: http://www.faro.com.ru/upload/FARO_Laser_Scanner_Focus_S_350.pdf

50. Faro Scene Datasheet [Электронный ресурс] / FARO Technologies, Inc. – 2019. – 2с. Режим доступа: http://www.faro.com.ru/upload/FARO_SCENE_RUS_.pdf
51. Sabbatini, L. Technological roadmap to boost the introduction of agvs in industrial applications / L. Sabbatini, ect. // ICCP 2013. – 2013. – Pp. 203–208.
52. Origin and OriginPro Datasheet [Электронный ресурс] / OriginLab Corporation. – 2017. Режим доступа: <https://www.originlab.com/index.aspx?go=PRODUCTS/Origin>
53. Sweep Visualizer Datasheet [Электронный ресурс] / Scanse LLC. – 2017. Режим доступа: <https://scanse.io/download/sweep-visualizer>
54. Sweep User Manual Datasheet [Электронный ресурс] / Scanse LLC. – 2017. – 17с. Режим доступа: https://s3.amazonaws.com/scanse/Sweep_user_manual.pdf
55. Самохин, И. Сканирующий лазерный дальномер [Электронный ресурс] / И. Самохин // ТехMedia. – 2016. Режим доступа: <https://habr.com/ru/post/393685/>
56. Agrawal, R. Драйвер сканирующего лазерного дальномера [Электронный ресурс] / R. Agrawal // GitHub – 2018. Режим доступа: https://github.com/rohbotics/xv_11_laser_driver
57. Zemax, оптика Fisheye [Электронный ресурс] / ZEMAX LLC URL.: – 2018. Режим доступа: <https://www.zemax.com/products/opticstudio>
58. 3D камера 3D-A5000 [Электронный ресурс] / «Малленом Системс». – 2018. Режим доступа: <https://www.mallenom.ru/products/mashinnoe-zrenie/3d-kamera-so-strochno-kadrovoj-razvertkoj-3d-a5000/#%D0%A1%D0%BF%D0%B5%D1%86%D0%B8%D1%84%D0%B8%D0%BA%D0%B0%D1%86%D0%B8%D1%8F>
59. Лю, В. Методы планирования пути в среде с препятствиями (обзор) / В. Лю // Математика и математическое моделирование. – 2018. – №01. С. 15–58.
60. Janet, J.A. The essential visibility graph: An approach to global motion planning for autonomous mobile robots / J.A. Janet, R.C. Luo, M.G. Kay // IEEE intern. conf. on robotics and automation. – 1995. – Vol.2. Pp. 1958–1963

61. Han-Pang, H. Dynamic visibility graph for path planning / H. Han-Pang, C. Shu-Yun // IROS 2004. – 2004. – Vol.3. Pp. 2813–2818.
62. Habib, M.K. Efficient method to generate collision free paths for an autonomous mobile robot based on new free space structuring approach / M.K. Habib, H. Asama // IROS 1991. – 1991. – Pp. 563–567.
63. Wallgrun, J.O. Voronoi graph matching for robot localization and mapping / J.O. Wallgrun // Transactions on computational science IX. – 2010. – Pp. 76–108.
64. Amato, N.M. A randomized roadmap method for path and manipulation planning / N.M. Amato, Y. Wu // IEEE intern. conf. on robotics and automation 1996. – 1996. – Vol.1. Pp. 113–120.
65. Kuffner, J.J. RRT-connect: An efficient approach to to single-query path planning / J.J. Kuffner, S.M. LaValle // ICRA 2000. – 2000. – Vol.2. Pp. 995–1001.
66. Elfes, A. Using occupancy grids for mobile robot perception and navigation / A. Elfes // Computer. – 1989. – Vol.22. No.6. Pp. 46–57.
67. Russell, S.J. Artificial intelligence: A modern approach. / S.J. Russell, P. Norvig. // 3rd ed. Upper Saddle River: Prentice Hall. – 2010. – pp. 1132-1133.
68. Ferguson, D. Using interpolation to improve path planning: The field D* algorithm / D. Ferguson, A. Stentz // J. of Field Robotics. – 2006. – Vol.23. no.2.– Pp. 79–101.
69. Daniel, K. Theta*: Any-angle path planning on grids / K. Daniel, A. Nash, S. Koenig S., A. Felner // J. of Artificial Intelligence Research. – 2010. – Vol.39. – Pp. 533–579.
70. Osher, S. Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations / S. Osher, J.A. Sethian // J. of Computational Physics. – 1988. – Vol.79. no.1. – Pp. 12–49.
71. Ge, S.S. New potential functions for mobile robot path planning / S.S. Ge, Y.J. Cui // IEEE Trans. on Robotics and Automation. – 2000. – Vol.16. No.5. – Pp. 615–620.

72. Xi-yong, Z. Virtual local target method for avoiding local minimum in potential field based robot navigation / Xi-yong Z., Jing Z. // *J. of Zhejiang Univ. - Science A.* – 2003. Vol.4. No.3. – Pp. 264–269.
73. Borenstein, J. The vector field histogram-fast obstacle avoidance for mobile robots / J. Borenstein, Y. Koren // *IEEE Trans. on Robotics and Automation.* – 1991. – Vol.7. No.3. – Pp. 278–288.
74. Fliess, M. Flatness and defect of non-linear systems: introductory theory and examples / M. Fliess, J. Levine, P. Martin, P. Rouchon // *Intern. J. of Control.* – 1995. Vol.61. No.6. – Pp. 1327–1361.
75. Culligan, K. Three-dimensional flight experiments using on-line mixed-integer linear programming trajectory optimization / K. Culligan, M. Valenti, Y. Kuwata, J.P. How // *ACC'2007.* – 2007. – Pp. 5322–5327.
76. Schouwenaars, T. Mixed integer programming for multi-vehicle path planning / T. Schouwenaars, B. De Moor, E. Feron, J. How // *ECC 2001.* – 2001. – Pp. 2603–2608.
77. Masehian, E. Motion planning and control of mobile robot using Linear Matrix Inequalities (LMIs) / E. Masehian, G. Habibi // *IROS 2007.* – 2007. – Pp. 4277–4282.
78. Colorni, A. Distributed Optimization by Ant Colonies / A. Colorni, M. Dorigo, V. Maniezzo // *Actes de la première conférence européenne sur la vie artificielle.* – 1991. – 134—142.
79. Glasius, R. Neural network dynamics for path planning and obstacle avoidance / R. Glasius, A. Komoda, Stan C.A.M. Gielen. // *Neural Networks.* – 1995. – Vol.8. No.1. – Pp. 125–133.
80. Мак-Каллок, У.С. Логическое исчисление идей, относящихся к нервной активности / У.С. Мак-Каллок, В. Питтс // *Автоматы: Сб.* – М.: Изд-во иностр. лит. – 1956. – С. 363–384.
81. Eberhart, R. A new optimizer using particle swarm theory / R. Eberhart, J. Kennedy // *MHS'95.* – 1995. – Pp. 39–43.

82. Egerstedt, M. Behavior based robotics using hybrid automata / M. Egerstedt // Hybrid HSCC 2000. – 2000. – Pp. 103–116
83. Ng, J. Performance comparison of bug navigation algorithms / J. Ng, T. Braunl // J. of Intelligent and Robotic Systems. – 2007. – Vol.50. No.1. – Pp. 73–84.
84. Yufka, A. Performance comparison of the BUG's algorithms for mobile robots / A. Yufka, O. Parlaktuna // Intern. symp. on INnovations in intelligent SYSTems and applications: INISTA 2009. – 2009. – Pp. 416–421.
85. ТИПОВАЯ ИНСТРУКЦИЯ ПО ОХРАНЕ ТРУДА ДЛЯ ВОДИТЕЛЕЙ АВТОПОГРУЗЧИКОВ. ТИП М-009-2000 [Электронный ресурс] // Правовая консультационная служба - 2019. Режим доступа: <http://www.zakonprost.ru/content/base/54652/>
86. Eckermann, E. Die Achsschenkellenkung und andere Fahrzeug-Lenksysteme. / E. Eckermann // YUMPU. – 2019. Режим доступа: <https://www.yumpu.com/de/document/read/3500484/die-achsschenkellenkung-und-andere-fahrzeug-lenksysteme>
87. GB-Patent 4212.27.01.1818. Improvements on axletrees applicable to four-wheeled carriages / R. Ackermann — 1818.
88. Чудаков, Е.А. Теория автомобиля. / Е.А. Чудаков — М.: Машгиз. — 1950. — 344 с.
89. Фаробин, Я.Е. Теория поворота транспортных машин / Я.Е. Фаробин — М.: Машиностроение. — 1970. — 176 с.
90. Раймпель, Й. Шасси автомобиля: Рулевое управление. Пер. с нем. В.Н. Пальянова. Под ред. А.А.Гальбрейха. / Й. Раймпель — М.: Машиностроение, 1987. — 232 с.
91. Tolubko, V.B. Decision of problem of manoeuvrability by an increase degrees of mobile of machines with the non-holonomic copulas of the wheeled type / V.B. Tolubko, Vasil'ev, B.G. Berezan A.M. // Mekhanika ta mashinobuduvannya, Kharkov Polytechnic Institute. — 2009. — No.2. — Pp. 96-102.

92. Титаренко, Д.Н. Базовые понятия геометрии ходовой части и рулевого управления. Учебное пособие [Электронное издание] / Д.Н. Титаренко — Спб.: АвтоСпециалист+. — 2012. — 31 с. Режим доступа: <https://docplayer.ru/28340429-Uchebnoe-posobie-bazovye-ponyatiya-geometrii-hodovoy-chasti-i-rulevogo-upravleniya.html>
93. Беляков, В.В. Расчет углов поворота управляемых колес автомобиля с учетом увода [Электронное издание] / В.В. Беляков, и др. // Труды НГТУ им. Р.Е. Алексеева. — 2019. — №2(125). — С. 156-162. Режим доступа: <https://cyberleninka.ru/article/n/raschet-uglov-povorota-upravlyaemyh-koles-avtomobilya-s-uchetom-uvoda>
94. Берестова, С.А. Кинематическое управление движением колесных транспортных средств / С.А. Берестова, Н.Е. Мисюра, Е.А. Митюшов // Вестник Удмуртского университета. Математика. Механика. Компьютерные науки. — 2015. — т.25. вып.2. — С. 254-266
95. Мисюра, Ф.Д. УПРАВЛЕНИЕ ДВИЖЕНИЕМ ЧЕТЫРЕХКОЛЕСНОГО ТРАНСПОРТНОГО СРЕДСТВА ПО ЗАДАННОЙ ТРАЕКТОРИИ [Электронное издание] / Ф.Д. Мисюра, Е.А. Митюшов // Материалы VII Международной студенческой научной конференции «Студенческий научный форум». — 2016. Режим доступа: <https://scienceforum.ru/2015/article/2015015214>
96. Фаробин, Я.Е. О рациональной форме рулевой трапеции / Я.Е. Фаробин — М.: Автомобильная промышленность. — 1959. — т.2 — С. 15-17.
97. Fiala, E. Kraftkorrigierte Lenkanggeometrie / E. Fiala // ATZ. — 1959. — №2. — С. 86-92.
98. Hasselgruber, H. Veerhaiten eines Kraftfahrzeuges bei Kurveenfahrt Automobiltechn Z / H. Hasselgruber // Selgruber. — 1965 — №7 — С. 43-50.
99. Алексеев, А.А. Выбор измерительных преобразователей для робототележки, функционирующей в недетерминированной среде. / А.А. Алексеев и др. // Инновации на основе информации-онных и телекоммуникационных технологий: Материалы научно-практической конференции. – М.: НИУ ВШЭ, 2015. С. 431-433.

100. Палагута, К.А. Использование объектно-ориентированного языка программирования для определения размеров объекта / К.А. Палагута, А.А. Алексеев, И.С. Шубникова // Материалы Международной научно-технической конференции «IN-TERMATIC – 2014», часть 5. МИРЭА, 2014 С. 84-85.
101. Glasius, R. Neural Network Dynamics for Path Planning and Obstacle Avoidance / R. Glasius, A. Komoda, S. Gielen // Neural Networks. – 1995. – № 8
102. Техническая документация VDI 2198, электротележка ET1 [Электронный ресурс] / АО "Балканкар Рекорд" — 2016. Режим доступа:http://balkancar-record.com/Content/images/platforms/specifications/LT_ET1_RU.pdf
103. Алексеев, А.А. Автоматический объезд препятствий внутрицеховым транспортом / А.А. Алексеев, и др. // Естественные и технические науки. – 2014. – №12 – С. 301-306. (Из списка ВАК)
104. Русак, А.В. Разработка алгоритмов управления ориентации мобильных роботов: диссертация канд. техн. наук: 29.05.2007 / Русак Елена Викторовна – Спб. Санкт-Петербургский государственный университет информационных технологий, мехатроники и оптики – 2007. – 150 с.
105. Cox, M.G. The Numerical Evaluation of B-Splines / M.G. Cox // National Physical Laboratory DNAC. – 1971. – Pp. 1-17.
106. Lyche, T. B-Splines and Spline Approximation [Электронный ресурс] / T. Lyche, C. Manni, H. Speleers // Department of Mathematics, University of Oslo, Norway. – 2003. – Pp. 1-39. Режим доступа: http://www.mat.uniroma2.it/~speleers/cime2017/material/notes_lyche.pdf
107. Rida, T. The Bernstein polynomial basis: A centennial retrospective [Электронный ресурс] / T. Rida Farouki // Computer Aided Geometric Design. – 2012. – Vol.29. – Pp. 379–419. Режим доступа: <http://mae.engr.ucdavis.edu/~farouki/bernstein.pdf>
108. Жермен-Лакур, П. Математика и САПР: В 2-х кн. Кн. 2; Пер. с франц. / П. Жермен-Лакур, П.Л. Жорж, Ф. Пистр, П. Безье — М.: Мир. — 1989. — 264 с.
109. Роджерс, Д. Математические основы машинной графики. / Д. Роджерс, Дж. Адамс — М.: Мир. — 2001. — с. 604.

110. Палагута, К.А. Алгоритм расчёта участка траектории для параллельной парковки автомобиля. / К.А. Палагута, А.А. Алексеев А.А. // Известия МГИУ. — 2009. — №4 (17). — с. 2-7.
111. Laumond, J.P. Robot Motion Planning and Control / J.P. Laumond, Lecture Notes in Control and Information Sciences. — 1998. — Vol. 229. — 343 p.
112. Буданов, В.М. О движении колесных роботов / В.М. Буданов, Е.А. Девянин // ПММ. — 2003. — Т. 67. вып. 2. — С. 244-255.
113. Громов, Ю.Ю. Специальные разделы теории управления. Оптимальное управление динамическими системами: учеб. пособие – 2-е изд., стереотип / Ю.Ю. Громов, и др. –:Тамб. Изд-во Тамб. гос. техн. ун-та. — 2007. — 108 с.
114. Карлова-Ильина, Е.В. Управление движением мобильного робота на базе нечёткой логики [Электронный ресурс] / Е.В. Карлова-Ильина. — 2012. Режим доступа: http://magazine.stankin.ru/arch/n_16/13/index.htm
115. Иванов, А.М. Основы конструкции автомобиля. / А.М. Иванов, и др. - М. ООО "Книжное издательство "За рулём". — 2005. — 336 с.
116. Острецов, А.В. Автомобильные подвески: Учебное пособие по дисциплине "Конструкция автомобиля и трактора" для студентов вузов, обучающихся по специальности 190201 (150100) "Автомобиле и тракторостроение". Часть I. / А.В. Острецов, П.А. Красавин, В.В. Воронин, Л.А. Павлова — М.: МГТУ "МАМИ". — 2011. — 162 с.
117. Кушвид, Р.П. Испытания автомобиля: учебник. / Р.П. Кушвид – М.: МГИУ. — 2011. – 351 с.
118. Алексеев, А.А. Управление траекторией движения, автоматизированного внутрицехового транспортного средства / А.А. Алексеев // Естественные и технические науки. — 2014. – №12. – С. 298-300. (Из списка ВАК)
119. Палагута, К.А. Поиск оптимальной опорной траектории автоматизированного транспортного средства / К.А. Палагута, А.А. Алексеев // Материалы 4-й Всероссийской мультikonференции. — 2011. — Том 2. — С. 3-8.
120. Палагута, К.А. Метод формирования области поиска для оптимизации траектории уклонения автомобиля от лобового столкновения / К.А. Палагута, А.А.

Алексеев // Мехатроника, Автоматизация, Управление. — 2012. — №2. — С. 44–48.

121. Никулин, Е.А. Компьютерная геометрия и алгоритмы машинной графики: Учебное пособие / Е.А. Никулин — Спб.: ВХБ – Петербург. — 2003. – 550с.

122. Судоплатов, С.В. Математическая логика и теория алгоритмов: Учебное пособие / С.В. Судоплатов, Е.В. Овчинникова, Новосибирский государственный технический университет (НГТУ). — 2008 – 224 с.

123. Гладков, Л.А. Генетические алгоритмы 2-е изд., испр. и доп. / Гладков Л.А., Курейчик В.В., Курейчик В.М. — М.: ФИЗМАТЛИТ, 2006 320 с

124. Емельянов В.В., Курейчик В.В., Курейчик В.М. Теория и практика эволюционного моделирования. — М.: ФИЗМАТЛИТ. — 2003. — 432с.

125. Рутковская, Д. Нейронные сети, генетические алгоритмы и нечеткие системы — 2-е изд. / Д. Рутковская, М. Пилиньский, Л. Рутковский — М: Горячая линия-Телеком. — 2008. — 452с.

126. Гладков, Л.А. Генетические алгоритмы 2-е изд., испр. и доп. / Л.А. Гладков, В.В. Курейчик, В.М. Курейчик — М.: ФИЗМАТЛИТ. — 2006. — 320с.

127. Алексеев, А.А. Метод оптимизации сложных кривых с использованием генетических методов поиска / А.А. Алексеев, К.А. Палагута // Информационные и телекоммуникационные технологии. — 2011. — №13. — С. 90-100. (Из списка ВАК)

128. Батищев, Д.И. Применение генетических алгоритмов к решению задач дискретной оптимизации. Учебно-методический материал по программе повышения квалификации «Информационные технологии и компьютерное моделирование в прикладной математике» [Электронный ресурс] / Д.И. Батищев, Е.А. Ней-марк, Н.В. Старостин — Нижний Новгород — 2007. - 85 с. Режим доступа: <http://www.unn.ru/pages/e-library/aids/2007/15.pdf>

129. Панченко, Т.В. Генетические алгоритмы: учебно-методическое пособие / Т.В. Панченко, под ред. Ю.Ю. Тарасевича. — Астрахань: Издательский дом «Астраханский университет» — 2007. — 87с.

130. Зеленин, С.Ф. Учебник по вождению автомобиля. / С.Ф. Зеленин — М.: ООО «Мир Автокниг». — 2006. — 80с.

131. Лосиный тест [Электронный ресурс] / Режим доступа: http://ru.wikipedia.org/wiki/Лосиный_тест

132. Рязанцев, В.Д. Большая политехническая энциклопедия. / В.Д. Рязанцев - М.: Мир и Образование — 2011. — 704 с.

Приложение А
Трассировки экспериментов «Выбор маршрута»
(Информационное)

Тестирование алгоритма выбора маршрута проводилось на трёх случайно сформированных выборках из исходного кодового пространства, представленного в таблице 4.8. Далее приведена полная пошаговая трассировка для эксперимента №1.

Таблица Б.1. – Исходная и финальная популяции решений

№	Исходная популяция		Финальная популяция	
	CD	F _ц	CD	F _ц
0	0001001	0,01546	0000010	0,01547
1	0001011	0,01544	0000010	0,01547
2	0001111	0,01542	0000010	0,01547
3	0010000	0,01543	0000010	0,01547
4	0010001	0,01544	0000010	0,01547
5	0100001	0,01543	0000010	0,01547
6	0100011	0,01544	0000010	0,01547
7	0100101	0,0154	0000010	0,01547
8	0101000	0,0154	0000010	0,01547
9	0110001	0,0154	0000010	0,01547
10	0110010	0,0154	0000010	0,01547
11	0111001	0,01538	0000010	0,01547
12	0111100	0,0154	0000010	0,01547
13	1000000	0,0154	0000010	0,01547

Здесь № – номер хромосомы в популяции, CD – кодовое слово, соответствующее хромосоме, F_ц – значение функции приспособленности хромосомы.

Таблица Б.2. – Адаптация исходной и финальной популяций решений

№	Исходная популяция		Финальная популяция	
	CD	P_s	CD	P_s
0	0001001	7,16271%	0000010	7,14286%
1	0001011	7,15345%	0000010	7,14286%
2	0001111	7,14418%	0000010	7,14286%
3	0010000	7,14881%	0000010	7,14286%
4	0010001	7,15345%	0000010	7,14286%
5	0100001	7,14881%	0000010	7,14286%
6	0100011	7,15345%	0000010	7,14286%
7	0100101	7,13491%	0000010	7,14286%
8	0101000	7,13491%	0000010	7,14286%
9	0110001	7,13491%	0000010	7,14286%
10	0110010	7,13491%	0000010	7,14286%
11	0111001	7,12565%	0000010	7,14286%
12	0111100	7,13491%	0000010	7,14286%
13	1000000	7,13491%	0000010	7,14286%

Здесь № – номер хромосомы в популяции, CD – кодовое слово, соответствующее хромосоме, P_c – вероятность скрещивания.

Таблица Б.3. – Операция скрещивания (кроссовер), итерация №1

№	Пара родителей		Пара потомков	
	CD ₁	CD ₂	CD ₁	CD ₂
1	001 0001	010 1000	0011000	0100001
2	01 10010	01 01000	0101000	0110010

3	0 101000	0 010000	0010000	0101000
4	0010 001	0101 000	0010000	0101001
5	0 010001	0 110001	0110001	0010001
6	01 10010	01 01000	0101000	0110010
7	0001001	0001001	0000000	1000000

Здесь № – номер пары хромосом, отобранных для операции кроссовера, CD_n – кодовое слово, соответствующее хромосомам родительского поколения и поколения потомков. Для родительского поколения символом «|» обозначена точка расщепления родительских хромосом во время операции кроссовера для каждого шага. Для таблиц Б.4 – Б.23 обозначения аналогичные.

Таблица Б.4. – Операция скрещивания (кроссовер), итерация №2

№	Пара родителей		Пара потомков	
	CD_1	CD_2	CD_1	CD_2
1	10000 00	01100 10	1000010	0110000
2	001100 0	010100 1	0011001	0101000
3	00110 00	01100 10	0011010	0110000
4	0101 000	0110 010	0101010	0110000
5	011 0010	011 0010	0110010	0110010
6	01010 00	10000 00	0101000	1000000
7	010 1000	010 1001	0101001	0101000

Таблица Б.5. – Операция скрещивания (кроссовер), итерация №3

№	Пара родителей		Пара потомков	
	CD_1	CD_2	CD_1	CD_2
1	0 110000	0 101000	0101000	0110000

2	01 01000	01 01001	0101001	0101000
3	010100 0	011001 0	0101000	0110010
4	010100 0	100001 0	0101000	1000010
5	0110 010	0011 010	0110010	0011010
6	0101 000	0101 001	0101001	0101000
7	0101010	0101010	0000010	1000010

Таблица Б.6. – Операция скрещивания (кроссовер), итерация №4

№	Пара родителей		Пара потомков	
	CD ₁	CD ₂	CD ₁	CD ₂
1	0101000	0101000	0000010	1000010
2	001101 0	010100 0	0011010	0101000
3	0101 000	0101 001	0101001	0101000
4	01010 00	01100 10	0101010	0110000
5	0101 000	0101 000	0101000	0101000
6	010100 0	010100 1	0101001	0101000
7	0 101000	0 101001	0101001	0101000

Таблица Б.7. – Операция скрещивания (кроссовер), итерация №5

№	Пара родителей		Пара потомков	
	CD ₁	CD ₂	CD ₁	CD ₂
1	0101001	0101001	0100010	1000010
2	010100 0	010101 0	0101000	0101010
3	1000010	1000010	1000010	0100010
4	0 101000	0 101001	0101001	0101000

5	1000010	1000010	1000010	0100010
6	0101 001	1000 010	0101010	1000001
7	0101 000	0101 000	0101000	0101000

Таблица В.7. – Операция скрещивания (кроссовер), итерация №6

№	Пара родителей		Пара потомков	
	CD ₁	CD ₂	CD ₁	CD ₂
1	0101000	0101000	0000010	1000010
2	0 101000	0 100010	0100010	0101000
3	1000001	1000001	1000001	0000001
4	10000 01	10000 10	1000010	1000001
5	010101 0	010100 0	0101010	0101000
6	0101010	0101010	0000001	1000001
7	01010 10	01010 00	0101000	0101010

Таблица Б.8. – Операция скрещивания (кроссовер), итерация №7

№	Пара родителей		Пара потомков	
	CD ₁	CD ₂	CD ₁	CD ₂
1	0101 000	0101 000	0101000	0101000
2	10000 01	01010 00	1000000	0101001
3	01 01000	00 00001	0100001	0001000
4	1 000001	1 000001	1000001	1000001
5	0101 000	0000 001	0101001	0000000
6	1 000010	1 000001	1000001	1000010
7	0000 010	1000 010	0000010	1000010

Таблица Б.9. – Операция скрещивания (кроссовер), итерация №8

№	Пара родителей		Пара потомков	
	CD ₁	CD ₂	CD ₁	CD ₂
1	10000 10	10000 01	1000001	1000010
2	100 0001	100 0001	1000001	1000001
3	0101000	0101000	0100001	1000001
4	0100001	0100001	0000001	1000001
5	1000010	1000010	1000010	0100010
6	1000 010	0101 001	1000001	0101010
7	100001 0	010100 0	1000010	0101000

Таблица Б.10. – Операция скрещивания (кроссовер), итерация №9

№	Пара родителей		Пара потомков	
	CD ₁	CD ₂	CD ₁	CD ₂
1	0000 001	0000 001	0000001	0000001
2	10000 01	10000 10	1000010	1000001
3	1000 001	0000 001	1000001	0000001
4	100 0001	100 0010	1000010	1000001
5	10 00010	00 00001	1000001	0000010
6	1 000001	1 000010	1000010	1000001
7	100000 1	000000 1	1000001	0000001

Таблица Б.11. – Операция скрещивания (кроссовер), итерация №10

№	Пара родителей		Пара потомков	
	CD ₁	CD ₂	CD ₁	CD ₂

1	0 000001	1 000010	0000010	1000001
2	0 000001	1 000010	0000010	1000001
3	0000001	0000001	0000000	1000010
4	1 000010	1 000010	1000010	1000010
5	10 00010	10 00010	1000010	1000010
6	0000 001	0000 001	0000001	0000001
7	00000 01	10000 01	0000001	1000001

Таблица Б.12. – Операция скрещивания (кроссовер), итерация №11

№	Пара родителей		Пара потомков	
	CD ₁	CD ₂	CD ₁	CD ₂
1	100000 1	000000 1	1000001	0000001
2	10 00010	00 00010	1000010	0000010
3	1 000001	0 000010	1000010	0000001
4	1 000001	0 000010	1000010	0000001
5	00000 01	00000 10	0000010	0000001
6	000 0001	100 0010	0000010	1000001
7	1000 001	0000 001	1000001	0000001

Таблица Б.13. – Операция скрещивания (кроссовер), итерация №12

№	Пара родителей		Пара потомков	
	CD ₁	CD ₂	CD ₁	CD ₂
1	0 000001	0 000010	0000010	0000001
2	1000 001	1000 010	1000010	1000001
3	000 0001	100 0010	0000010	1000001

4	0000 010	0000 010	0000010	0000010
5	0000 010	1000 010	0000010	1000010
6	000 0010	100 0001	0000001	1000010
7	00 00010	00 00010	0000010	0000010

Таблица Б.14. – Операция скрещивания (кроссовер), итерация №13

№	Пара родителей		Пара потомков	
	CD ₁	CD ₂	CD ₁	CD ₂
1	000 0010	000 0010	0000010	0000010
2	00000 10	00000 10	0000010	0000010
3	00 00010	00 00010	0000010	0000010
4	00 00001	10 00010	0000010	1000001
5	0000 010	0000 010	0000010	0000010
6	100 0010	000 0010	1000010	0000010
7	10000 01	00000 10	1000010	0000001

Таблица Б.15. – Операция скрещивания (кроссовер), итерация №14

№	Пара родителей		Пара потомков	
	CD ₁	CD ₂	CD ₁	CD ₂
1	00000 10	00000 10	0000010	0000010
2	0 000010	0 000010	0000010	0000010
3	1 000001	0 000010	1000010	0000001
4	00000 10	00000 10	0000010	0000010
5	0000 010	0000 010	0000010	0000010
6	000 0010	100 0001	0000001	1000010

7	0 000010	0 000010	0000010	0000010
---	----------	----------	---------	---------

Таблица Б.16. – Операция скрещивания (кроссовер), итерация №15

№	Пара родителей		Пара потомков	
	CD ₁	CD ₂	CD ₁	CD ₂
1	000 0010	100 0010	0000010	1000010
2	000001 0	000001 0	0000010	0000010
3	00 00001	00 00010	0000010	0000001
4	000 0010	000 0010	0000010	0000010
5	0000 010	1000 010	0000010	1000010
6	000001 0	000001 0	0000010	0000010
7	00 00010	00 00010	0000010	0000010

Таблица Б.17. – Операция скрещивания (кроссовер), итерация №16

№	Пара родителей		Пара потомков	
	CD ₁	CD ₂	CD ₁	CD ₂
1	000 0010	000 0010	0000010	0000010
2	0000 010	0000 010	0000010	0000010
3	000001 0	000001 0	0000010	0000010
4	000 0010	000 0010	0000010	0000010
5	00000 10	00000 10	0000010	0000010
6	00 00010	00 00010	0000010	0000010
7	000 0010	000 0010	0000010	0000010

Таблица Б.18. – Операция скрещивания (кроссовер), итерация №17

№	Пара родителей	Пара потомков
---	----------------	---------------

	CD ₁	CD ₂	CD ₁	CD ₂
1	0000 010	0000 010	0000010	0000010
2	00000 10	00000 10	0000010	0000010
3	0000 010	0000 010	0000010	0000010
4	00 00010	00 00010	0000010	0000010
5	000 0010	000 0010	0000010	0000010
6	0000 010	0000 010	0000010	0000010
7	0 000010	0 000010	0000010	0000010

Таблица Б.19. – Операция скрещивания (кроссовер), итерация №18

№	Пара родителей		Пара потомков	
	CD ₁	CD ₂	CD ₁	CD ₂
1	00000 10	00000 10	0000010	0000010
2	0000 010	0000 010	0000010	0000010
3	00 00010	00 00010	0000010	0000010
4	00000 10	00000 10	0000010	0000010
5	000 0010	000 0010	0000010	0000010
6	0 000010	0 000010	0000010	0000010
7	0000 010	0000 010	0000010	0000010

Таблица Б.20. – Операция скрещивания (кроссовер), итерация №19

№	Пара родителей		Пара потомков	
	CD ₁	CD ₂	CD ₁	CD ₂
1	0 000010	0 000010	0000010	0000010
2	0000 010	0000 010	0000010	0000010
3	000 0010	000 0010	0000010	0000010

4	0 000010	0 000010	0000010	0000010
5	00000 10	00000 10	0000010	0000010
6	000001 0	000001 0	0000010	0000010
7	0000 010	0000 010	0000010	0000010

Таблица Б.21. – Операция скрещивания (кроссовер), итерация №20

№	Пара родителей		Пара потомков	
	CD1	CD2	CD1	CD2
1	000001 0	000001 0	0000010	0000010
2	000 0010	000 0010	0000010	0000010
3	0 000010	0 000010	0000010	0000010
4	00 00010	00 00010	0000010	0000010
5	000001 0	000001 0	0000010	0000010
6	000 0010	000 0010	0000010	0000010
7	00 00010	00 00010	0000010	0000010

Таблица Б.22. – Операция скрещивания (кроссовер), итерация №21

№	Пара родителей		Пара потомков	
	CD ₁	CD ₂	CD ₁	CD ₂
1	000 0010	000 0010	0000010	0000010
2	000001 0	000001 0	0000010	0000010
3	00 00010	00 00010	0000010	0000010
4	000001 0	000001 0	0000010	0000010
5	00000 10	00000 10	0000010	0000010
6	00 00010	00 00010	0000010	0000010
7	0000 010	0000 010	0000010	0000010

Таблица Б.23. – Операция скрещивания (кроссовер), итерация №22

№	Пара родителей		Пара потомков	
	CD ₁	CD ₂	CD ₁	CD ₂
1	000001 0	000001 0	0000010	0000010
2	00 00010	00 00010	0000010	0000010
3	000 0010	000 0010	0000010	0000010
4	000 0010	000 0010	0000010	0000010
5	00000 10	00000 10	0000010	0000010
6	00000 10	00000 10	0000010	0000010
7	0 000010	0 000010	0000010	0000010

Таблица Б.24. – Изменение значения целевой для каждой итерации

№	F _{цс}	F _{цп}	CD
1	0,01542	0,0154	0001001
2	0,01542	0,0154	0000000
3	0,01541	0,0154	0011010
4	0,01541	0,01542	0000010
5	0,01541	0,0154	0000010
6	0,01542	0,0154	0100010
7	0,01543	0,01542	0000001
8	0,01542	0,01542	0000000
9	0,01542	0,0154	0000001
10	0,01544	0,01548	0000001
11	0,01544	0,01541	0000000
12	0,01545	0,01548	0000001

13	0,01545	0,01547	0000001
14	0,01546	0,01548	0000001
15	0,01546	0,01547	0000001
16	0,01546	0,01547	0000001
17	0,01547	0,01547	0000010
18	0,01547	0,01547	0000010
19	0,01547	0,01547	0000010
20	0,01547	0,01547	0000010
21	0,01547	0,01547	0000010
22	0,01547	0,01547	0000010

Здесь № – номер итерации алгоритма, $F_{\text{ср}}$ – среднее значение целевой функции для поколения решений, $F_{\text{пл}}$ – лучшее значение целевой функции для поколения, CD – кодовое слово лучшего решения в поколении.

На рисунке Б.1 показан отобранный маршрут из имеющихся альтернатив.

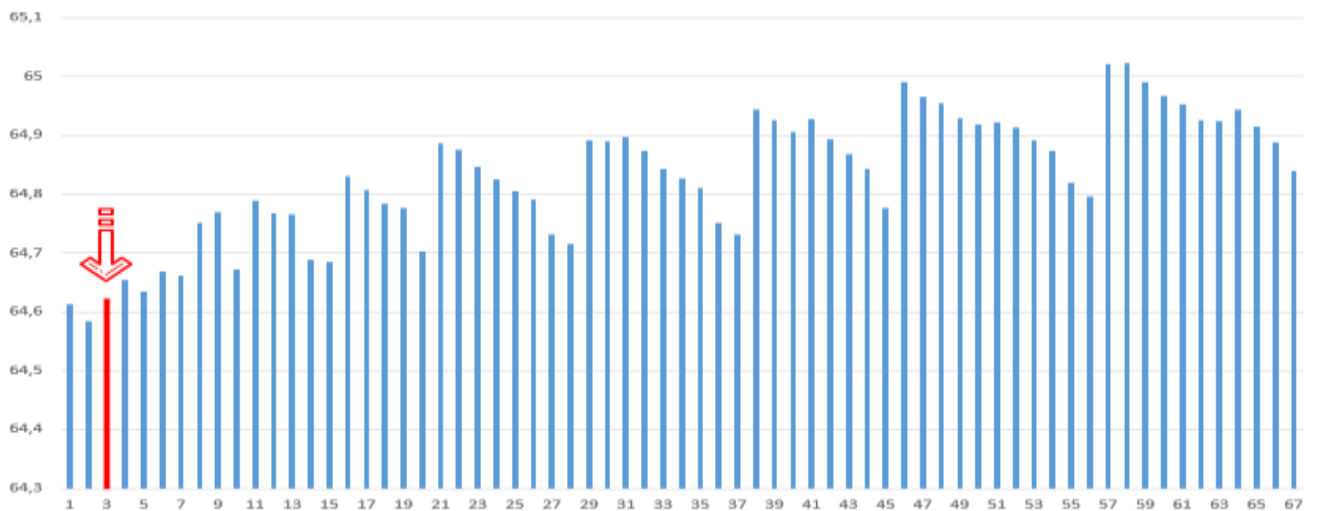


Рисунок Б.1. – Отобранный маршрут. На оси абсцисс указаны номера маршрутов пространства поиска, на оси ординат указана протяжённость маршрута. Отобранный маршрут показан стрелкой.

На рисунке Б.2. проиллюстрирован процесс приближения к решению.

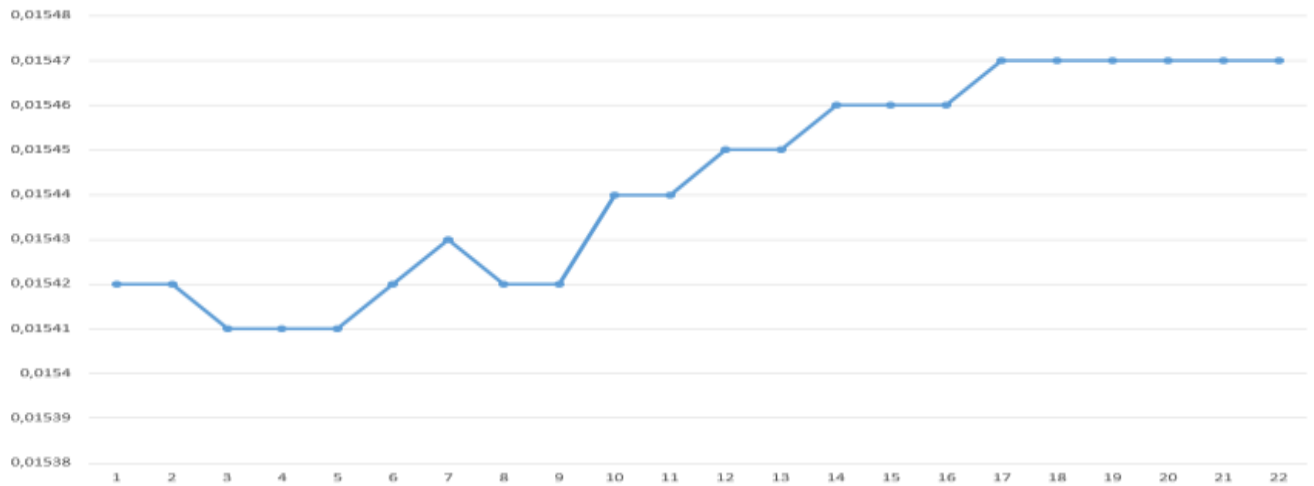


Рисунок Б.2. – Приближение к решению. На оси абсцисс указаны номера итерации алгоритма, на оси ординат средние для поколения значения функции приспособленности.

Приложение Б

Обоснование применения генетического алгоритма к задаче выбора маршрута из множества альтернативных маршрутов.

(Информационное)

Применение генетических алгоритмов в задачах выбора маршрута из множества альтернатив является распространённым решением [101]. Что обусловлено проблемой формализации задачи выбора маршрута. Так, например, для постановки задачи поиска глобального минимума/максимума необходимо иметь аналитическое представление области поиска.

В данном конкретном случае возможность получить универсальное аналитическое представление области поиска затруднительно в силу того, что набор альтернативных маршрутов одновременно зависит от:

- метода формирования множества, на котором осуществляется поиск;
- направления движения робота;
- расстояния до препятствия;
- взаимного расположения робота и препятствия;
- скорости сближения/опережения.

Фактически для каждого задания область поиска маршрута будет иметь свой «профиль» (огибающая пики столбовой диаграммы, рисунки В.1 – В.3).

Так, например, на рисунке В.1 показан профиль области поиска для задания, подробно разобранный в работе (выбор маршрута – эксперимент №1). На рисунке В.2 отображён профиль области поиска того же задания при условии, что робот движется задним ходом. Рисунок В.3 демонстрирует то же задание при условии модификации механизма формирования задания.

Причём в зависимости от состояния окружающей робот среды может изменяться не только профиль, но и размерность области поиска.

Например, опережение препятствия движущегося со направленно с роботом.



Рисунок В.1. – Профиль области поиска маршрута. Движение робота вперёд. На оси абсцисс указаны номера маршрутов, составляющих область поиска, на оси ординат – протяжённости маршрутов.



Рисунок В.2. – Профиль области поиска маршрута. Движение робота задним ходом.



Рисунок В.3. – Профиль области поиска маршрута. Изменены настройки модуля ГТ.

Следует заметить, что при этом остаётся возможность выбора маршрута, имеющего наименьшую протяжённость формально, а именно, путём последовательного перебора всех альтернатив.

На рисунке В.4. показан график сравнения быстродействий генетического алгоритма и метода прямого перебора (при проведении контрольного испытания, в обоих случаях использовался одинаковый метод вычисления длин кривых).

Из рисунка В.4 видно, что генетический алгоритм эффективнее работает с пространствами поиска большого размера. Данный факт обусловлен механизмом действия генетических алгоритмов, так как расчёт длин всех кривых, образующих пространство поиска, не производится.

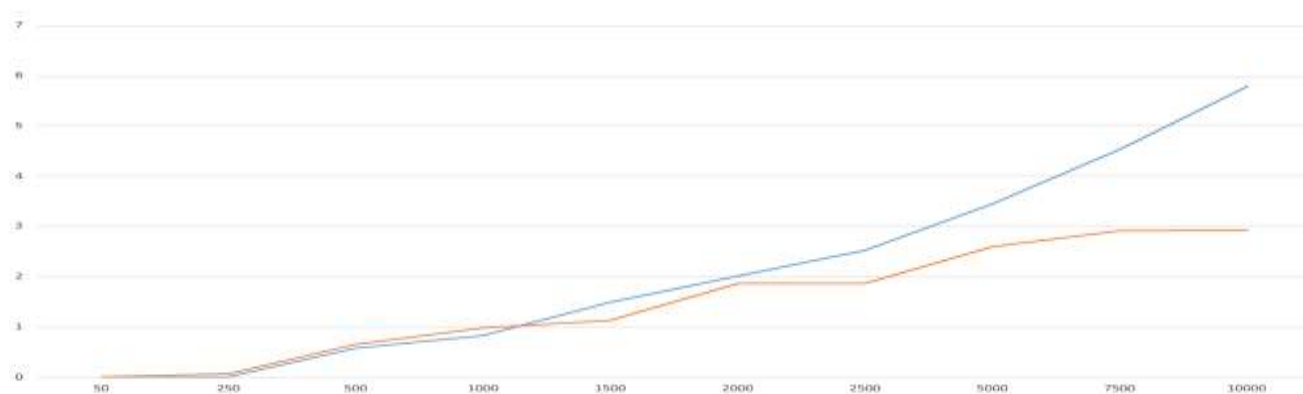


Рисунок В.4. – Сравнение быстродействия генетического алгоритма и метода прямого перебора всех имеющихся альтернатив. На оси абсцисс указано количество кривых в области поиска, на оси ординат – время в секундах.

На рисунках В.5 и В.6 показаны результаты исследования стабильности поведения использованного генетического алгоритма при многократном (100 экспериментов) перезапуске. Перезапуск осуществлялся на одном и том же множестве решений. Повторяющийся характер выборок не является ошибкой, так как обусловлен особенностью реализацией механизма выборки случайных значений функцией `rand` языка программирования, привязанной к частоте тактирования процессора. По результатам исследования установлено, что среднее значение количества итераций для данной выборки составляет 17.11, среднее время 0.062с. Так как скорость движения робота по маршруту не превышает 2,777 м/с, то отсюда следует, что за время «принятия решения» робот сместится не более чем на 0,172 метра, что является допустимым с точки зрения технических регламентов, определяющих правила транспортировки объектов хранения на складских предприятиях.



Рисунок В.5. – Исследование поведения ГА на фиксированном множестве решений при множественном перезапуске. На оси абсцисс указан номер эксперимента, на оси ординат – количество итераций, выполненных в процессе поиска решения.



Рисунок В.6. – Исследование поведения ГА на фиксированном множестве решений при множественном перезапуске. На оси абсцисс указан номер эксперимента, на оси ординат – время процесса поиска решения.

Приложение В

Листинги программ

(Информационное)

Далее приведены листинги программ, для описания объектов использован язык PHP5, последующая реализация осуществлялась на Ddelphi 7:

- Листинг Г.1– Реализация генетического алгоритма, файл: GeneticAlgorithm.php

Листинг Г.1. – Реализация генетического алгоритма

```

/**
 * Генетический алгоритм
 */
class GeneticAlgorithm {
/**
 * @var array Bezier3 кривые Безье
 */
private $curves = [];
/**
 * @var array кодовое пространство
 */
private $codeSpace = [];
/**
 * @var integer длина кодового слова
 */
private $codeLength = 0;
/**
 * var integer Кол-во особей в популяции
 */
private $populationSize = 0;
/**
 * @var максимальное кол-во итераций
 */
private $iterationStop = 0;

```

```
/**
 * @var integer счётчик итераций
 */
private $iterationCounter = 0;
/**
 * @var array результат работы алгоритма
 */
private $result = [];
/**
 * @var integer время начала поиска
 */
private $timer = 0;
/**
 * @var array кэш значений целевых функций
 */
private $cache = [];
/**
 * @var флаг отладки
 */
private $debug = false;
/**
 * @var integer вероятность скрещивания внутри пары хромосом
 */
private $probabilityCrossover = 80;
/**
 * @var integer вероятность мутации гена в хромосоме
 */
private $probabilityMutation = 2;
/**
 * @var array буфер вывода
 */
private $outputBuffer = [];
/**
 * @var array условие останова
 */
private $stopConditions = [
    'current' => [
        'middle' => 0,
        'max' => 0,
```

```

        'iteration' => 0,
    ],
    'previous' => [
        'middle' => 0,
        'max' => 0
    ],
    'maxIteration' => 5
];
/**
 * @var array массив статистики по экспериментам
 * @static
 */
private $statistics = [];
/**
 * Конструктор
 *
 * @param $curves массив кривых Безье
 * @param $populationSize кол-во особей в популяции
 * @param $iterationStop кол-во итераций после которых следует принуди-
тельное завершение
 * @return GeneticAlgorithm
 */
public function __construct($curves, $populationSize=14, $itera-
tionStop=100)
{
    // Сохранение массива кривых
    $this->curves = $curves;
    $this->populationSize = (int)$populationSize;
    $this->iterationStop = (int)$iterationStop;

    // Проводим проверку того, что кол-во особей в популяции кратно двум
    if ($this->populationSize % 2) {
        $this->populationSize++;
    }

    // Двоично десятичное кодирование
    $this->setCodeSpace();
}
/**

```

```

* Запускает алгоритм
*
* @param void
* @return void
*/
public function search()
{
    // Получение базовой популяции
    $basePopulation = $this->getBasePopulation();

    // Очистка от предыдущего исполнения
    $this->iterationCounter = 0;
    $this->cache = [];
    $this->result = [];
    $this->outputBuffer = [];
    $this->stopConditions = [
        'current' => [
            'middle' => 0,
            'max' => 0,
            'iteration' => 0,
        ],
        'previous' => [
            'middle' => 0,
            'max' => 0
        ],
        'maxIteration' => 5
    ];
    $this->timer = microtime();

    // Сохранение данных в буфере вывода
    $this->outputBuffer['populations'] = [];

    if ($this->debug) {
        self::__print_r('Стартовая выборка');
        self::__print_r($basePopulation);
    }

    // Запуск алгоритма поиска. В случае прерывания срабатывает
    // блок определения лучшего потомка

```

```

try {
    $this->calculate($basePopulation);
} catch (Exception $e) {
    // Вывод результатов
    if ($this->debug) {
        self::__print_r("Результат");
        self::__print_r($this->outputBuffer);
    }
}
return $this->output();
}
/**
 * Выводит кодовое пространство
 *
 * @param void
 * @return string
 */
public function getCodeSpace()
{
    $html = '';
    if ($this->debug) {
        $this->__print_r($this->codeSpace);
    }
    ob_start();
    require 'template-searchSpace.phtml';
    $html = ob_get_contents();
    ob_end_clean();

    return $html;
}
/**
 * Основной метод. Осуществляет рекурсивный поиск решения
 *
 * @param void
 * @return void
 */
private function calculate($population)
{
    // Накопитель данных

```

```

$buffer = [];

if ($this->debug) {
    self::__print_r("Популяция №{$this->iterationCounter}");
    __print_r($population);
}

// Проверка условий остонова
$searchStop = $this->isComplete();
if ($searchStop) {
    $this->result['time'] = microtime() - $this->timer;
    $this->result['iterationTotal'] = $this->iterationCounter;
    $this->result['population'] = $population;
    $this->result['bestChild'] = $this->getBestChild($population);
    $this->outputBuffer['result'] = $this->result;
    if ($this->debug) {
        self::__print_r($this->result);
    }
    // Сохранение статистики
    $this->statistics[] = $this->result;
    throw new Exception('Stop');
}

// Вычисления адаптации текущего поколения
$chromosomes = $this->getPopulationAdaptation($population, $buffer);
//self::__print_r($buffer);
//exit();
// Вычисления сектора рулетки соответствующего каждому потомку
$chromosomesSectors = $this->getWheelSectors($chromosomes);
// Определеие родительского пула хромосом
$parentPool = $this->getParrentPool($chromosomesSectors, $population);
// Выбор пар для скрещивания
$loversPool = $this->getLoversPool($parentPool, $buffer);
// Получение нового поколения
$newPopulation = $this->getNewPopulation($loversPool, $buffer);
if ($this->debug) {
    self::__print_r("Новое поколение");
    self::__print_r($newPopulation);
}

```

```

    // Переход к следующей итерации
    $this->outputBuffer['populations'][] = $buffer;
    $this->iterationCounter++;
    $this->calculate($newPopulation);
}
/**
 * Выбирает лучшего потомка популяции
 *
 * @param array $population популяция
 * @return string
 */
private function getBestChild($population)
{
    // Вычисления адаптации текущего поколения
    $bestValue = 0;
    $bestChild = null;
    $chromosomes = $this->getPopulationAdaptation($population);
    foreach ($chromosomes as $childNumber => $childValue) {
        if ($bestValue < $childValue) {
            $bestChild = $population[$childNumber];
        }
    }
    return $bestChild;
}
/**
 * Возвращает true если процесс поиска завершён и false в противном случае
 *
 * @param void
 * @return boolean
 */
private function isComplete()
{
    if ($this->debug) {
        self::__print_r("Проверка стагнации");
        self::__print_r($this->stopConditions);
    }
    // Алгоритм вошёл с стагнацию

```

```

        if ($this->stopConditions['maxIteration'] == $this->stopCondi-
tions['current']['iteration']) {
            return true;
        }
        // Предел эксперимента достигнут естественным путём
        if ($this->iterationCounter >= $this->iterationStop) {
            return true;
        }
        return false;
    }
/**
 * Определение нового поколения
 *
 * @param array $loversPool массив пар родительских хромосом
 * @param $buffer array накопитель данных
 * @return array новое поколение
 */
private function getNewPopulation($loversPool, &$buffer)
{
    if ($this->debug) {
        self::__print_r("Определение нового поколения решений");
    }
    $newPopulation = [];
    $buffer['crossover'] = [];
    foreach ($loversPool as $lovers) {
        $childs = [];
        // Определение произошло ли скрещивание у текущей пары. Если скре-
щивание
        // не произошло пара полностью без изменения переносится в новый
пул решений
        $crossover = (rand(0, 100) <= $this->probabilityCrossover) ? true
: false;
        if (true) {
            // Операция скрещивания
            $childs = $this->crossover($lovers, $buffer);
            // Мутация (не реализована)
        } else { // Родители перебрались в следующее поколение
            $childs = $lovers;
        }
    }
}

```

```

        $buffer['crossover'][] = array('parents'=>$childs,
'childs'=>$childs);
    }
    $newPopulation = array_merge($newPopulation, $childs);
}

if ($this->debug) {
    self::__print_r("Полученный дочерний пул:");
    self::__print_r($newPopulation);
}
$buffer['newPopulation'] = $newPopulation;
return $newPopulation;
}
/**
 * Скрещивание хромосом
 *
 * @param array $parents пара родительских хромосом
 * @param $buffer array накопитель данных
 * @return array пара дочерних хромосом
 */
private function crossover($parents, &$buffer)
{
    $bufferCrossover = [];
    $bufferExplode = [];

    $childs = [];
    $explodePoint = rand(1, $this->codeLength - 1);

    if ($this->debug) {
        self::__print_r("Процесс скрещивания, точка газрыва генотипа:
{$explodePoint}, родительская пара:");
        self::__print_r($parents);
    }

    $bufferCrossover['parents'] = $parents;

    foreach ($parents as $parent) {
        $childs[] = [
            substr($parent, 0, $explodePoint),

```

```

        substr($parent, $explodePoint)
    ];

}

if ($this->debug) {
    self::__print_r("Разорванные пары:");
    self::__print_r($childs);
}

$bufferExplode = $childs;

$childs = [
    $childs[0][0] . $childs[1][1],
    $childs[1][0] . $childs[0][1]
];

// Если в кодовом пространстве хромосомы с полученным кодом отсутствуют,
то переносим
// в дочернее поколение родительскую хромосому
if (!array_search($childs[0], $this->codeSpace)) {
    $childs[0] = $parents[0];
    $bufferExplode = null;
}

if (!array_search($childs[1], $this->codeSpace)) {
    $childs[1] = $parents[1];
    $bufferExplode = null;
}

if ($this->debug) {
    self::__print_r("Полученные потомки:");
    self::__print_r($childs);
}

$bufferCrossover['childs'] = $childs;
if ($bufferExplode) {
    $bufferCrossover['crossover'] = $bufferExplode;
}
$buffer['crossover'][] = $bufferCrossover;

return $childs;
}

```

```

/**
 * Определяет пары хромосом которые будут скрещиваться между собой
 *
 * @param $parentPool array родительский пул
 * @param $buffer array накопитель данных
 * @param array пары хромосом для скрещивания
 */
private function getLoversPool($parentPool, &$buffer)
{
    $loversTotal = $this->populationSize / 2;
    $loversCounter = 0;
    $loversPool = [];
    while ($loversCounter < $loversTotal) {
        $lovers = array_rand($parentPool, 2);
        foreach ($lovers as &$lover) {
            $lover = $parentPool[$lover];
        }
        $loversPool[] = $lovers;
        $loversCounter++;
    }

    if ($this->debug) {
        self::__print_r("Пары для скрещивания");
        self::__print_r($loversPool);
    }

    $buffer['loversPool'] = $loversPool;
    return $loversPool;
}
/**
 * Определения родительского пула хромосом
 *
 * @param $chromosomes array распределение секторов колеса рулетки
 * @param $population array коды хромосом текущей популяции
 * @return array коды родительского пула
 */
private function getParrentPool($chromosomes, $population)
{
    if ($this->debug) {

```

```

        self::__print_r('Формирование родительского пула хромосом');
    }

    $parentPool = [];
    $parentCounter = 0;
    while ($parentCounter < $this->populationSize) {
        $sectorValue = rand(0, 100);
        if ($this->debug) {
            self::__print_r("На рулетке выбрано значение:
{$sectorValue}");
        }
        $parentPool[] = $population[$this->findChromosome($chromosomes,
$sectorValue)];
        $parentCounter++;
    }
    if ($this->debug) {
        self::__print_r("Родительский пул:");
        self::__print_r($parentPool);
    }
    return $parentPool;
}
/**
 * Определяет номер элемента массива в популяции соответствующий
 * значению выпавшему на колесе рулетки $sectorValue на основании
 * секторов принадлежащих хромосомам $chromosomes
 *
 * @param $chromosomes array массив секторов хромосом
 * @param $sectorValue integer значение на выпавшее на колесе рулетки
 * @return integer
 */
private function findChromosome($chromosomes, $sectorValue)
{
    foreach ($chromosomes as $code => $sector) {
        if (($sector[0] <= $sectorValue) && ($sector[1] > $sectorValue)) {
            return $code;
        }
    }
}
/**

```

```

* Определяет сектор рулетки соответствующий каждой
* хромосоме в наборе $chromosomes
*
* @param array $chromosomes список вероятностей скрещивания для каждой
хромосомы в поколении
* @return array
*/
private function getWheelSectors($chromosomes)
{
    // Распределение секторов колеса рулетки
    $sectorStart = 0;
    $currentSector = 0;
    foreach ($chromosomes as &$chromosomeSector) {
        $currentSector = [$sectorStart, $sectorStart + $chromosomeSector];
        $sectorStart += $chromosomeSector;
        $chromosomeSector = $currentSector;
    }

    // Обеспечиваем гарантированный полный охват рулетки
    $chromosomes[count($chromosomes) - 1][1] = 101;

    if ($this->debug) {
        self::__print_r("Распределение секторов рулетки между хромосо-
мами");
        self::__print_r($chromosomes);
    }

    return $chromosomes;
}
/**
* Определение адаптации текущей популяции $population
*
* @param $population array массив популяции решений
* @param $buffer array накопитель данных
* @return array массив хромосом
*/
private function getPopulationAdaptation($population, &$buffer=null)
{
    if ($this->debug) {

```

```

        self::__print_r("Вычисление адаптации");
    }

    // Оценка всех хромосом популяции
    $chromosomes = [];
    $populationAdaptation = 0;
    if (!is_null($buffer)) {
        $buffer['population'] = $population;
    }

    foreach ($population as $childCode) {
        $chromosome = $this->getTargetFunctionValue($childCode);
        $populationAdaptation += $chromosome;
        $chromosomes[] = $chromosome;
    }
    if ($this->debug) {
        self::__print_r("Суммарное значение адаптации поколения = {$populationAdaptation}, одоптации особей:");
        self::__print_r($chromosomes);
    }

    if (!is_null($buffer)) {
        $buffer['chromosomes'] = $chromosomes;
    }

    // Обновление условий останова
    $this->updateStopConditions($chromosomes, $population, $populationAdaptation, $buffer);

    // Расчёт вероятности скрещивания
    foreach ($chromosomes as &$chromosomeAdaptation) {
        $chromosomeAdaptation = round((($chromosomeAdaptation / $populationAdaptation) * 100), 5);
    }
    if (!is_null($buffer)) {
        $buffer['adaptaion'] = $chromosomes;
    }

    if ($this->debug) {

```

```

    $wheelSumm = 0;
    foreach ($chromosomes as $value) {
        $wheelSumm += $value;
    }
    self::__print_r("Сумма секторов колеса рулетки {$wheelSumm}.
Вероятности скрещивания:");
    self::__print_r($chromosomes);
}

return $chromosomes;
}
/**
 * Обновляет условия принудительного останова алгоритма
 *
 * @param $chromosomes массив значений функции приспособленности поколения
 * @param $population массив хромосом поколения
 * @param $populationAdaptation суммарное значения функции адаптации
 */
private function updateStopConditions($chromosomes, $population, $popula-
tionAdaptation, &$buffer)
{
    // Получение среднего и лучшего значения функции адаптации для поколе-
ния
    $adaptaionMiddle = round(($populationAdaptation / $this->popula-
tionSize), 5);
    $adaptaionMax = 0;
    foreach ($chromosomes as $chromosomeNumber => $chromosomesAdaptation)
    {
        if ($chromosomesAdaptation > $adaptaionBest) {
            $adaptaionMax = $chromosomesAdaptation;
        }
    }

    if ($this->debug) {
        self::__print_r("Обновление условий останова");
        self::__print_r("Среднее значение адаптации для поколения:
{$adaptaionMiddle}");
        self::__print_r("Лучшее значение адоптации: {$adaptaionMax}");
    }
}

```

```

if ($this->stopConditions['previous']['middle'] == $adaptaionMiddle) {
    $this->stopConditions['current']['iteration']++;
} else {
    $this->stopConditions['current']['iteration'] = 0;
}
// Обновление значений
$this->stopConditions['previous']['middle'] = $adaptaionMiddle;
$this->stopConditions['previous']['max'] = $adaptaionMax;
$this->stopConditions['current']['middle'] = $adaptaionMiddle;
$this->stopConditions['current']['max'] = $adaptaionMax;

$buffer['adaptaionMiddle'] = $adaptaionMiddle;
$buffer['adaptaionMax'] = $adaptaionMax;

if ($this->debug) {
    self::__print_r($this->stopConditions);
}
}
/**
 * Вычисляет значение функции приспособленности для кривой с кодом $code
 *
 * @param $code string
 * @return float
 */
private function getTargetFunctionValue($code) {
    if (empty($this->cache[$code])) {
        $curveName = array_search($code, $this->codeSpace);
        if (!$curveName) {
            exit("<p>Кривая с кодом {$code} отсутствует в кодовом простран-
стве!</p>");
        }
        $curve = $this->curves[$curveName];
        $curve->getLinesLength();
        $this->cache[$code] = round (1 / $curve->_line_len, 5);
    }
    return $this->cache[$code];
}
/**

```

```

* Формирует начальную популяцию. Выбирает из кодового пространства
* $populationSize потомков.
*
* @param void
* @return array базовая популяция
*/
private function getBasePopulation()
{
    $basePopulation = array_rand($this->codeSpace, $this->populationSize);
    foreach ($basePopulation as &$child) {
        $child = $this->codeSpace[$child];
    }
    return $basePopulation;
}
/**
* Формирует непрерывное кодовое пространство, устанавливает $codeSpace и
$codeLength
*
* @param void
* @return void
*/
private function setCodeSpace()
{
    $this->codeLength = 0;
    $this->codeSpace = [];

    // Получение кода всех кривых и максимальной длины кодового слова
    $curvesCounter = 0;
    foreach ($this->curves as $curveID => $curve) {
        $code = decbin($curvesCounter);
        $codeLength = strlen($code);

        if ($codeLength > $this->codeLength) {
            $this->codeLength = $codeLength;
        }
        $this->codeSpace[$curveID] = decbin($curvesCounter);

        $curvesCounter++;
    }
}

```

```

// Выравнивание длины всех кодовых последовательностей
$this->codeLengthNormalization();

// Если работа в режиме отладки, выводим кодовое пространство и длину
КОДОВЫХ СЛОВ
if ($this->debug) {
    self::__print_r($this->codeSpace);
    self::__print_r($this->codeLength);
}
}
/**
 * Нормализация длины кодовых слов. Устанавливает для всех кодовых слов
одинаковую
 * длину, дополняя короткие слова нулями с права.
 *
 * @param void
 * @return void
 */
private function codeLengthNormalization() {
    foreach ($this->codeSpace as &$code) {
        if (strlen($code) < $this->codeLength) {
            while (strlen($code) < $this->codeLength) {
                $code = '0' . $code;
            }
        }
    }
}
/**
 * Выводит содержимое буфера в виде требуемых таблиц
 *
 * @param void
 * @return void
 */
private function output()
{
    $html = '';
    if ($this->debug) {
        $this->__print_r($this->outputBuffer);
    }
}

```

```

    }
    ob_start();
    require 'template-GeneticAlgorithm.phtml';
    $html = ob_get_contents();
    ob_end_clean();

    return $html;
}
/**
 * Статистика по массиву экспериментов
 *
 * @param void
 * @return string
 */
public function getStatistic()
{
    $html = '';
    if ($this->debug) {
        $this->__print_r($this->statistics);
    }
    ob_start();
    require 'template-GeneticAlgorithmStatistics.phtml';
    $html = ob_get_contents();
    ob_end_clean();

    return $html;
}
/**
 * Служебный принтер
 *
 * @var $v mixed
 * @return void
 */
public static function __print_r($v)
{
    echo '<pre>';
    print_r($v);
    echo '<pre>';
}

```