

ГОСУДАРСТВЕННЫЙ НАУЧНЫЙ ЦЕНТР РОССИЙСКОЙ ФЕДЕРАЦИИ –
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ УНИТАРНОЕ ПРЕДПРИЯТИЕ
«Центральный ордена Трудового Красного Знамени научно-исследовательский
автомобильный и автотранспортный институт «НАМИ»
ГНЦ РФ ФГУП «НАМИ»

На правах рукописи



БЕЛЯКОВ Федор Юрьевич

**АЛГОРИТМ ГЕНЕРАЦИИ ЛОКАЛЬНОЙ ТРАЕКТОРИИ ДВИЖЕНИЯ
АВТОМОБИЛЯ С УЧЕТОМ ДОРОЖНЫХ ОГРАНИЧЕНИЙ ПО
МАРШРУТУ ДВИЖЕНИЯ**

Специальность 2.5.11. Наземные транспортно–технологические средства и
комплексы

ДИССЕРТАЦИЯ

на соискание ученой степени кандидата технических наук

Научный руководитель:
кандидат физико-математических наук
Евграфов Владимир Владимирович

Москва – 2026

Содержание

Введение	5
ГЛАВА 1. Обзор отечественных и зарубежных разработок по методам генерации траектории движения беспилотного автомобиля	19
1.1 Базовые задачи движения беспилотного автомобиля	22
1.2 Планирование траектории	24
1.2.1 Диаграмма Вороного	25
1.2.2 Метод потенциалов	27
1.2.3 Сетка занятости	29
1.2.4 Метод случайных деревьев	30
1.3 Локализация и картография	31
1.4 Управление движением	32
1.5 Преимущества использования квадратичного программирования для реального времени	34
1.6 Сравнение с другими подходами	35
1.6.1 Ограничения существующих и предлагаемого подходов при генерации локальной траектории движения автомобиля	37
1.7 Формулировка задачи квадратичного программирования в общем виде	38
1.8 Использование координат Френе	39
1.9 Выводы	42
1.10 Цель и задачи исследования	44
ГЛАВА 2. Математическая модель автомобиля и алгоритмы следования траектории	47
2.1 Модель автомобиля в пространстве состояний	48
2.2 Линеаризация с помощью ряда Тейлора	50
2.3 Математические модели автомобиля	53
2.3.1 Кинематическая велосипедная модель	53
2.3.2 Динамическая велосипедная модель	55
2.3.3 Линеаризованная модель поперечной динамики и ошибок слежения	57
2.3.4 Кинематическая велосипедная модель в фрейме Френе	60

2.4	Аппроксимация опорной траектории кубическим В-сплайном . . .	63
2.5	Алгоритмы следования траектории	64
2.5.1	Принцип работы алгоритма Чистое преследование (Pure Pursuit, PP)	66
2.5.2	Принцип работы линейно-квадратичного регулятора следования траектории	68
2.5.3	Принцип работы алгоритма модельно-предиктивного управления (Model Predictive Control, MPC)	71
2.5.4	Преобразование задачи MPC к задаче квадратичного программирования (QP)	76
2.5.5	Результаты моделирования и воспроизводимость вычислительного эксперимента	80
2.6	Выводы	93
ГЛАВА 3. Разработка алгоритма планирования траектории с учетом дорожной обстановки		95
3.1	Алгоритм сглаживания траектории движения беспилотного транспортного средства на основе квадратичной оптимизации без учета свободного пространства	99
3.1.1	Постановка задачи сглаживания траектории без учета свободного пространства	99
3.1.2	Формат представления траектории в декартовой системе координат и дискретная модель	100
3.1.3	Предварительная аппроксимация опорной траектории В-сплайном	101
3.1.4	Целевая функция задачи квадратичного программирования	102
3.1.5	Ограничения и линеаризация динамики	102
3.1.6	Результаты моделирования и обсуждение влияния весов . .	103
3.2	Алгоритм сглаживания траектории движения беспилотного транспортного средства на основе квадратичной оптимизации с учетом ограничений на границы дороги	105
3.2.1	Формулировка задачи сглаживания траектории с учетом ограничений	105
3.2.2	Представление траектории и формирование ограничений на свободное пространство	105

3.2.3	Целевая функция задачи квадратичного программирования	108
3.2.4	Ограничения и линеаризация динамики	109
3.2.5	Численный эксперимент и обсуждение	109
3.3	Эффективное планирование траектории для автономных транспортных средств с использованием выпуклой оптимизации в системе координат Френе с учетом границ дороги	110
3.3.1	Представление в системе координат Френе	111
3.3.2	Формирование полигона проходимости и учет геометрии автомобиля	113
3.3.3	Формирование функционала качества	125
3.3.4	Результаты моделирования	126
3.3.5	Формирование коридора движения с отдельным учетом границ дороги и статических препятствий	133
3.3.6	Ускорение расчетов за счет фиксированного горизонта при решении QP методом OSQP	135
3.4	Выводы	139
ГЛАВА 4.	Экспериментальные исследования разработанных алгоритмов движения на физическом автомобиле	142
4.1	Испытательная платформа и архитектура системы	142
4.2	Протокол испытаний и условия проведения	144
4.2.1	Требования к условиям проведения	144
4.3	Виртуальная верификация в симуляторе Panosim	144
4.4	Проведение тестовых испытаний в городской среде	148
4.4.1	Сравнение с альтернативными подходами и вклад разработанного алгоритма	156
4.5	Выводы	158
	ОСНОВНЫЕ РЕЗУЛЬТАТЫ И ВЫВОДЫ ПО РАБОТЕ	159
	СПИСОК СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ	163
	СПИСОК ЛИТЕРАТУРЫ	164
	ПРИЛОЖЕНИЕ А. ПРОГРАММНАЯ РЕАЛИЗАЦИЯ АЛГОРИТМА ПОСТРОЕНИЯ ОГРАНИЧЕНИЙ ДЛЯ QP	172
	ПРИЛОЖЕНИЕ Б. АКТЫ О ВНЕДРЕНИИ	178

Введение

Актуальность темы исследования

Автоматизация управления автомобилем рассматривается в настоящее время как один из ключевых путей повышения безопасности дорожного движения и эффективности функционирования транспортной системы. По данным Министерства транспорта США, около 94% серьезных дорожно-транспортных происшествий обусловлены ошибками водителей [1]. В Российской Федерации в 2023 году около 88% ДТП произошло по вине водителей, что подчеркивает высокую значимость человеческого фактора в обеспечении безопасности движения [2]. Автономные транспортные системы (высокоавтоматизированные и беспилотные автомобили) позволяют существенно снизить влияние человеческого фактора и обеспечивают значимое уменьшение аварийности и числа жертв в дорожно-транспортных происшествиях. Исследованиями в США сделан прогноз о том, что широкое внедрение автономного транспорта к 2050 году может привести к экономическому эффекту до 800 млрд долларов США в год за счет уменьшения числа аварий, сокращения заторов и повышения производительности труда [3].

Мировая практика организации дорожного движения ориентирована на повышение эффективности использования улично-дорожной сети. В этом контексте автоматизированные системы управления движением позволяют оптимизировать режимы разгона и торможения, оптимальные величины скоростей и дистанций, в том числе на поворотах и при совершении маневров, что способствует снижению заторов и повышению пропускной способности дорожной инфраструктуры.

Алгоритмическая гибкость решений, применяемых в системах автономного вождения, обуславливает возможность переноса разработанных методов в смежные области. Подходы к планированию траектории и управлению движением, отработанные на автомобильном транспорте, могут быть адаптированы для мобильной робототехники, логистики, сельского хозяйства, горнодобывающей промышленности и эксплуатации техники в условиях, опасных для человека (ликвидация последствий чрезвычайных ситуаций и др.). Таким образом, исследования в области беспилотного

транспорта обладают межотраслевым характером и имеют высокую практическую значимость.

При этом повышение уровня автоматизации транспортных средств предъявляет жесткие требования к качеству выполнения траектории движения. Для обеспечения безопасности и комфорта необходимо, чтобы траектория была динамически реализуемой с точки зрения модели автомобиля, обеспечивала допустимые значения кривизны, поперечного ускорения и угловых скоростей, а также гарантировала выдерживание заданных границ полосы движения и безопасных расстояний до препятствий. Формирование таких траекторий с помощью систем управления движением транспортных средств в реальном времени в условиях сложной дорожной обстановки является нетривиальной научно-технической задачей.

В диссертационной работе рассматривается задача локального планирования и исполнения траектории движения беспилотного автомобиля в условиях городской дорожной среды. На вход задачи поступают: геометрия полосы движения, положение и габариты статических препятствий, текущее состояние автомобиля и опорная линия движения. Требуется формировать и исполнять траекторию, которая одновременно: не выходит за границы допустимого коридора движения, обеспечивает безопасный зазор до препятствий, удовлетворяет ограничениям на кривизну и скорость ее изменения, вычисляется за время, допустимое для работы бортовой системы управления в реальном времени.

Краткая характеристика развития систем автоматического управления движением

В России основателем исследований в области автоматических систем управления движением автомобилей (АСУД) является проф. А.А. Юрчевский, сформировавший в 1975 году новое направление в теории автомобиля, связанное с внедрением в автомобиль систем автоматического управления движением на основе средств технического зрения [4]. В последующем значимый вклад в развитие теории и практики автоматизации управления движением автомобилей внесли отечественные специалисты: С.В. Бахмутов [5], М.В. Нагайцев [6, 7], В.В. Евграфов [8], И.А. Куликов [9, 10], П.А. Васин [11], К.С. Яковлев [12], А.А. Андрейчук [13], Д.В. Ендачев [14], а также научные коллективы ФГУП «НАМИ», ФИЦ ИУ РАН, ведущих технических

университетов и промышленных компаний.

Идея создания полностью автономного автомобиля была сформулирована несколько десятилетий назад и прошла путь от концепций, близких к научной фантастике, до экспериментальных и опытно-промышленных образцов. Уже в 1980–1990-е годы в Европе был реализован проект EUREKA PROMETHEUS, в рамках которого под руководством Э. Дикманнса разработаны экспериментальные автомобили VaMP и VITA-2, продемонстрировавшие принципиальную возможность автономного движения по многополосным автострадам с выполнением перестроений и обгонов. В дальнейшем существенный вклад в развитие автономного транспорта внесли работы Steven M. LaValle, Claire J. Tomlin, Sebastian Thrun, Seth Teller и других зарубежных исследователей, занимавшихся задачами планирования движения, восприятия и управления [15—18].

В настоящее время в мировой и отечественной практике создан широкий спектр систем частичной и высокой автоматизации управления движением транспортных средств (ТС), включая продвинутые системы помощи водителю (Advanced Driver-Assistance Systems, ADAS), системы автоматической парковки, адаптивного круиз-контроля, удержания в полосе, роботакси и др., реализуемых как автомобильными концернами, так и технологическими компаниями. Конкретные алгоритмы планирования и генерации траектории в промышленных системах в большинстве случаев разработчиками и производителями систем не раскрываются и являются их коммерческой тайной.

Важно также отметить, что практика эксплуатации систем помощи водителю (СПВ) показывает, что эффективность работы систем разных производителей значительно отличается от заявленной. Функциональные и конструктивные недостатки зарубежных систем СПВ, неблагоприятные условия эксплуатации, особенно в Российской Федерации, становятся причиной потери эффективности их работы, что может приводить к дорожно-транспортным происшествиям.

Проведенные испытания показали [19], что прямое применение систем, алгоритмов управления движением и программного обеспечения зарубежных СПВ не обеспечивает необходимой высокой точности по маршруту движения автомобилей, эффективности и надежности их работы.

В частности, в работе [19] рассматривались типовые сценарии

столкновений при различных скоростных режимах и в условиях, характерных для эксплуатации в Российской Федерации: изношенная или отсутствующая разметка, загрязнение лобового стекла, осадки, туман и недостаточная освещенность. Показано, что фактическая эффективность серийных систем активной безопасности в условиях, близких к идеальным, достигает лишь порядка 57% предотвращенных типовых столкновений, тогда как потенциально для полнофункционального комплекса в сопоставимых условиях может быть достигнуто до 97%. Эти данные количественно подтверждают разрыв между заявленной и достигаемой на практике эффективностью и надежностью, а также указывают на риски деградации точности следования маршруту.

Изложенное подчеркивает актуальность дальнейшего проведения академических исследований, направленных на разработку формализованных и воспроизводимых методов генерации траекторий.

Степень проработанности темы исследования.

Задача планирования и генерации траектории движения беспилотного автомобиля, как правило, решается с использованием одного или комбинации нескольких классов методов: дискретных (решеточных и графовых), методов на основе выборки (sampling-based), оптимизационных методов, а также методов, основанных на обучении [20].

Методы дискретного поиска путей, такие как алгоритмы A^* , D^* и их модификации, включая метод планирования по безопасным временным интервалам (Safe Interval Path Planning), основаны на дискретизации конфигурационного пространства либо пространства состояний [21—23]. Указанные методы обеспечивают надежное построение маршрута, исключающего столкновения, в том числе в условиях сложной статической и динамической обстановки. Их преимуществом является развитая и хорошо формализованная теоретическая база, а также возможность строгого учета геометрических ограничений, накладываемых препятствиями окружающей среды. Вместе с тем, траектории движения, формируемые с использованием дискретных поисковых алгоритмов, как правило, имеют кусочно-линейный либо кусочно-гладкий характер и требуют последующей процедуры сглаживания для обеспечения кинематической и динамической реализуемости [24]. Ограничения на кривизну траектории, поперечные ускорения и показатели комфорта движения в подобных методах учитываются неявно либо вводятся в

приближенной форме. При использовании решеточных представлений пространства состояний учет габаритов транспортного средства и точных границ полосы движения зачастую реализуется посредством искусственного расширения препятствий, что приводит к избыточному консерватизму получаемых решений и, как следствие, снижению пропускной способности транспортного потока.

Методы планирования движения, основанные на стохастическом выборе конфигураций (методы на основе выборки), такие как алгоритмы быстро исследующих случайных деревьев, получили широкое распространение при решении задач поиска траекторий в высокоразмерных конфигурационных пространствах и в средах со сложной геометрией проходимой области [25, 26]. К данной группе относятся, в частности, алгоритмы RRT, RRT*, а также их модификации, ориентированные на двунаправленный поиск [27]. Основным достоинством указанных методов является обеспечение вероятностной полноты, то есть гарантии нахождения допустимой траектории при достаточно плотной выборке пространства состояний [25, 27]. Вместе с тем данные алгоритмы, как правило, не обеспечивают требуемого уровня гладкости траектории и не позволяют напрямую учитывать ограничения, связанные с управляемостью и динамикой автомобиля. В результате получаемые траектории характеризуются изломами и резкими изменениями кривизны, что затрудняет их непосредственное использование в системах управления движением. Учет динамических ограничений и реалистической кинематической модели транспортного средства в рамках методов на основе выборки существенно усложняет постановку задачи и, как правило, приводит к значительному росту вычислительных затрат [28]. По этой причине траектории, синтезируемые такими планировщиками, в большинстве практических систем автономного вождения требуют дополнительной переработки с использованием методов сглаживания или оптимизации, ориентированных на повышение их физической реализуемости и качества управления [28].

Оптимизационные методы планирования траектории движения транспортного средства, включая методы модельно-предиктивного управления и постановки задач квадратичного программирования, позволяют явно учитывать в целевой функции и системе ограничений требования к плавности движения, кривизне траектории, величине поперечного ускорения, а также динамические

свойства автомобиля [28]. Подобные подходы получили широкое распространение в системах автоматического управления движением, поскольку обеспечивают формализацию требований комфорта, управляемости и безопасности в виде строгих математических критериев [29]. Распространенным при этом является представление траектории движения в виде аппроксимации полиномиальными функциями либо сплайнами, в том числе В-сплайнами, что обеспечивает высокую гладкость траектории и непрерывность ее производных по пространственной координате [30]. Указанные свойства положительно сказываются на комфорте движения и уровне динамической нагруженности элементов ходовой части автомобиля. Однако при использовании подобных параметризаций контроль кривизны траектории и расстояния до границ полосы движения, как правило, осуществляется косвенным образом — через положение контрольных точек или коэффициенты аппроксимирующих функций [31]. В результате возможны локальные превышения допустимых значений кривизны и поперечного ускорения, а также снижение точности выдерживания габаритов транспортного средства относительно дорожной разметки, особенно в условиях сложной геометрии дорожного коридора или наличия жестких пространственных ограничений.

В последние годы активно развиваются методы, основанные на обучении (обучение с учителем, обучение с подкреплением, имитационное обучение), а также гибридные подходы, сочетающие классические алгоритмы поиска и оптимизации с обучаемыми компонентами. В этом направлении можно отметить работы по алгоритмам POLAMP (Policy Optimization that Learns Adaptive Motion Primitives) [32], FFStreams (Fast Search with Streams for Autonomous Maneuver Planning) [33], а также исследования К.С. Яковлева и его коллег по конфликтно-ориентированному и интервальному планированию движения групп мобильных агентов, в частности методы Conflict-Based Search и его модификации, обеспечивающие разрешение пространственно-временных конфликтов при координации множества агентов в общем рабочем пространстве [34—36]. Данные методы демонстрируют высокую адаптивность к сложной и динамически изменяющейся обстановке, однако требуют значительных объемов обучающих данных, сложны для формальной верификации и зачастую не обеспечивают строгих гарантий по кривизне траектории движения ТС, комфортности и соблюдению габаритных ограничений в общем виде.

Проведенный анализ показывает, что существующие подходы к генерации траектории движения беспилотного автомобиля обладают следующими существенными недостатками и ограничениями в контексте задач, рассматриваемых в диссертационной работе:

- недостаточно строгий учет геометрических и динамических ограничений автомобиля (кривизна траектории движения, поперечное ускорение, габариты) при гарантированном удержании в границах полосы движения;
- необходимость многоступенчатой обработки (поиск пути, сглаживание, последующая проверка) без единой формализованной выпуклой постановки, что усложняет анализ и настройку алгоритмов;
- ограниченные возможности по явному учету свободного пространства и статичных препятствий в рамках решения единой задачи оптимизации, пригодной для реализации в режиме, близком к реальному времени;
- недостаточная апробация алгоритмов в условиях, приближенных к реальной эксплуатации, включающих как имитационные испытания, так и дорожные испытания на физическом автомобиле.

Указанные недостатки и ограничения современных систем управления движением беспилотных транспортных средств (БТС) определяют направления, по которым целесообразно развивать методы генерации траекторий, которые лежат в основе задач, решаемых в настоящей диссертационной работе.

Выбор в пользу выпуклой оптимизации в координатах Френе обусловлен тем, что данный подход позволяет объединить в одной постановке требования, которые в известных схемах часто решаются разрозненно: геометрические ограничения дороги, ограничения на управляемость и требования к гладкости траектории. По сравнению с дискретным поиском и *sampling-based* методами, предлагаемый подход уменьшает объем постобработки и обеспечивает более предсказуемое время расчета. По сравнению с общими нелинейными постановками, он дает более устойчивую вычислительную процедуру для бортового применения за счет решения задач квадратичного программирования (QP).

Выбор направления исследования также обусловлен инженерной задачей внедрения алгоритма в реальный контур АСУД автомобиля *Lada Vesta*, где

критичны одновременно безопасность, воспроизводимость поведения алгоритма и ограниченный вычислительный бюджет.

При практической реализации существующих подходов типично проявляются следующие ограничения: для графовых и решеточных методов — зависимость качества от дискретизации и необходимость дополнительного сглаживания; для *sampling-based* методов — вариативность вычислительного времени и сложность явного учета динамических ограничений; для общих нелинейных оптимизационных методов — высокая вычислительная стоимость и чувствительность к начальному приближению.

Предлагаемый в работе подход также имеет ограничения применимости: предполагается достоверная локализация и корректная геометрия дорожного коридора; в базовой постановке учитываются преимущественно статические препятствия; используемая линеаризованная модель корректна в режимах, не требующих явного учета срыва сцепления и интенсивного бокового скольжения; при расширении горизонта и числа ограничений возрастает размерность *QP*-задачи, что требует дополнительной настройки и оптимизации вычислений.

В работе рассматривается движение в условиях, не приводящих к деградации восприятия и сцепных свойств покрытия (сухой/умеренно влажный асфальтобетон). Изменение коэффициента сцепления, осадки, загрязнение сенсоров и иные факторы ухудшения видимости относятся к отдельной задаче обеспечения функционирования подсистемы восприятия и в настоящей диссертации не исследуются.

Целью работы является повышение безопасности и качества автоматического движения автомобиля за счет разработки метода и реализующего его алгоритма синтеза локальной траектории движения автомобиля на основе выпуклого квадратичного программирования, обеспечивающих формирование в режиме реального времени гладкой и геометрически допустимой траектории с учетом границ проезжей части, статических препятствий, габаритов автомобиля, ограничений на кривизну и ее производную по дуговой координате.

Предлагаемый алгоритм направлен на обеспечение движения в пределах заданной полосы с возможностью маневра безопасного перестроения, на

снижение изменений кривизны траектории и поддержание комфортных динамических параметров движения БТС. Построение траектории движения осуществляется на основе параметризованной кинематической модели автомобиля и формулируется как задача выпуклой оптимизации, допускающей эффективное численное решение в рамках ограниченного вычислительного бюджета, соответствующего требованиям к бортовым системам управления. Разрабатываемое решение должно обеспечивать адаптацию движения БТС к криволинейной геометрии дороги, учитывать габариты автомобиля без избыточного увеличения вычислительной нагрузки системы управления движением и демонстрировать устойчивость и применимость при эксплуатации в условиях городской дорожной среды. Для достижения поставленной цели необходимо решить следующие **задачи**:

1. Провести анализ и классификацию методов и алгоритмов локального планирования и управления траекторным движением высокоавтоматизированных и беспилотных автомобилей; установить ограничения известных подходов при учете геометрии проезжей части, габаритов автомобиля, статических препятствий и динамических ограничений; сформулировать требования к разрабатываемому методу.

2. Обосновать выбор и выполнить адаптацию математических моделей автоматизированного управления движением автомобиля для решения задач локального планирования и траекторного управления в декартовой системе координат и системе координат Френе; сформировать систему геометрических, кинематических и динамических ограничений.

3. Разработать метод и алгоритм синтеза закона траекторного управления автомобилем на основе оптимального управления; сформировать критерий качества, учитывающий точность следования заданной траектории, интенсивность и плавность рулевого воздействия, а также ограничения исполнительного механизма и поперечной динамики автомобиля; выполнить сравнительное исследование законов управления в единых условиях моделирования.

4. Разработать метод и алгоритм синтеза локальной траектории движения автомобиля на основе выпуклой квадратичной оптимизации, включающий формирование допустимого коридора движения с учетом границ проезжей

части, статических препятствий и габаритов автомобиля, а также ограничения на кривизну траектории и скорость ее изменения; исследовать влияние параметров функционала и горизонта планирования на качество и вычислительную реализуемость решения.

5. Реализовать разработанные методы и алгоритмы его применения в составе программно-алгоритмического комплекса автоматизированной системы управления движением и провести их имитационную и натурную верификацию на автомобиле «Лада Веста» по показателям точности следования траектории, геометрической и динамической реализуемости, безопасности объезда препятствий и времени вычисления.

Объектом исследования является автомобиль «Лада Веста», оснащенный экспериментальной системой автоматического управления движением, включающей разработанные алгоритмы объезда препятствий в полосе движения с учетом текущей дорожной обстановки.

Методы исследования

Теоретические исследования основаны на методах теории автомобиля, теории автоматического управления, теории оптимального управления, математического моделирования, теории оптимизации и вычислительной математики. Экспериментальные исследования основаны на методах проведения вычислительных экспериментов и дорожных испытаний автомобилей, а также на методах обработки результатов испытаний.

Научная новизна работы:

1. Усовершенствован метод сглаживания опорной траектории движения автомобиля в декартовой системе координат на основе квадратичного программирования, по сравнению с аналогами отличающийся совместной параметрической минимизацией отклонения от опорной линии, кривизны траектории и скорости ее изменения, что позволяет формировать гладкое начальное приближение с заданным соотношением между точностью следования и динамической реализуемостью траектории.

2. Разработан метод формирования допустимого коридора движения автомобиля по данным о границах проезжей части и статических препятствиях, по сравнению с аналогами отличающийся применением карты расстояний и габаритной аппроксимации автомобиля для построения связного выпуклого полигона проходимости и его представления в виде системы линейных ограничений задачи квадратичного программирования.

3. Разработаны метод и алгоритм синтеза локальной траектории движения автомобиля в системе координат Френе в форме выпуклой задачи квадратичного программирования, по сравнению с аналогами отличающийся совместным явным учетом геометрии проезжей части, габаритов автомобиля, ограничений на кривизну и скорость ее изменения, а также применением функционала качества, согласующего точность следования, плавность траектории, соблюдение безопасных интервалов. Реализация в разработанном алгоритме фиксированного горизонта планирования обеспечивает неизменность структуры матриц задачи и предсказуемость времени вычисления.

Теоретическая значимость работы

Теоретическая значимость работы заключается в развитии научно-методического аппарата локального планирования движения высокоавтоматизированных и беспилотных автомобилей. Сформирована математическая постановка синтеза локальной траектории в системе координат Френе в виде выпуклой задачи квадратичного программирования, обеспечивающая совместный учет границ проезжей части, статических препятствий, габаритов автомобиля, ограничений на кривизну траектории и скорость ее изменения. Разработанные научные и методические положения устанавливают взаимосвязь между формированием допустимого коридора движения и построением гладкой, геометрически и кинематически допустимой траектории и могут быть использованы при дальнейшем развитии методов автоматического управления движением колесных транспортных средств.

Практическая значимость работы

1. Разработана алгоритмическая схема многоступенчатой генерации траектории, которая предназначена для использования в составе АСУД как типовой шаблон интеграции планировщика, обеспечивающего их воспроизводимую настройку, тестирование и перенос на иные аппаратные платформы.
2. Разработано программное обеспечение, реализующее функцию управления траекторией движения высокоавтоматизированного и беспилотного автомобиля для отслеживания референсной траектории.
3. Разработано программное обеспечение, реализующее оптимальный способ генерации траектории движения высокоавтоматизированного и беспилотного автомобиля с учетом окружающей дорожной обстановки.

Личный вклад автора состоит в том, что представленные результаты исследования получены автором и при его непосредственном участии. Автором самостоятельно разработаны методы сглаживания опорной траектории, формирования допустимого коридора и синтеза локальной траектории движения автомобиля, алгоритмическая схема многоступенчатой генерации траектории; разработаны и отлажены алгоритмы генерации и следования траектории. Осуществлена программная реализация разработанных алгоритмов; выполнены обработка и анализ данных натурных испытаний; сформулированы выводы и рекомендации по совершенствованию способов генерации траектории для объезда статичных препятствий в полосе движения высокоавтоматизированного и беспилотного автомобиля.

Реализация результатов работы

1. Результаты исследования используются во ФГУП «НАМИ» в центре «Интеллектуальные системы» при проведении исследований и разработке систем автоматического управления движением автомобилей.

2. Результаты исследования используются ООО «Инновационный центр КАМАЗ» для построения траекторий при цифровом моделировании пространственного движения перспективных колесных машин, а также отладке алгоритмов управления системой активной безопасности.

На защиту выносятся:

1. Метод сглаживания опорной траектории автомобиля на основе квадратичного программирования с параметрической балансировкой точности следования, кривизны и скорости ее изменения.
2. Метод формирования допустимого коридора движения с учетом границ проезжей части, статических препятствий и габаритов автомобиля.
3. Метод и алгоритм синтеза локальной траектории движения автомобиля в системе координат Френе на основе выпуклой квадратичной оптимизации с дорожными, габаритными и кинематическими ограничениями.
4. Результаты математического моделирования, имитационных и дорожных испытаний, подтверждающие геометрическую и динамическую реализуемость формируемых траекторий, безопасность объезда статических препятствий и вычислительную применимость разработанного метода в режиме реального времени.

Степень достоверности результатов работы

Достоверность математического моделирования движения высокоавтоматизированного и беспилотного автомобиля и результатов исследования разработанных алгоритмов управления движением подтверждается научной обоснованностью использованных методов, а также сопоставлением расчетных данных с измерениями, полученными в ходе дорожных испытаний автомобиля.

Достоверность результатов исследования разработанных алгоритмов подтверждается научной обоснованностью использованных методов, а также вычислением оценочных параметров качества выполнения маневров автомобилем и сопоставлением полученных параметров с результатами других исследований.

Апробация работы

Основные результаты работы докладывались на конференциях: 120-я международная научно-техническая конференция «Современные задачи и перспективы развития наземных транспортно-технологических машин и комплексов» 23–24 сентября 2024 года ФГБОУ ВО «Санкт-Петербургский государственный архитектурно-строительный университет», доклад: ”Алгоритм сглаживания траектории движения беспилотного транспортного средства на основе квадратичной оптимизации”; Международный автомобильный научный Форум МАНФ 2024 «Научно-технологическое развитие автомобильной промышленности Российской Федерации» 16–17 октября 2024// доклад: ”Алгоритм сглаживания траектории движения беспилотного транспортного средства на основе квадратичной оптимизации с учетом проходимого свободного пространства”.

Публикации

По теме диссертации опубликовано 9 печатных работ, из них 3 в изданиях, рекомендованных ВАК РФ для публикаций материалов диссертаций на соискание ученых степеней доктора и кандидата технических наук (перечень ВАК, база Scopus, база Web of Science), а также 5 свидетельств об официальной регистрации программ для ЭВМ.

Структура и объем работы

Диссертация состоит из введения, 4 глав, общих результатов и выводов, списка литературы и двух приложений. Общий объем работы составляет 171 страницу без приложений, включая 47 рисунков и 22 таблицы. Библиография работы содержит 75 наименований.

ГЛАВА 1. Обзор отечественных и зарубежных разработок по методам генерации траектории движения беспилотного автомобиля

Автономные транспортные средства стремительно развиваются в последние годы и охватывают различные области, включая робототехнику, информатику и инженерию. Разработка автономных транспортных средств может значительно сократить количество аварий по вине невнимательности / усталости водителя. Следует отметить, что особое значение для безопасного движения автомобиля имеет наличие надежного и адаптируемого планирования, особенно в условиях смешанного городского движения, когда на дорогах одновременно передвигаются автономные автомобили и управляемые человеком. Принятие критически важных решений, таких как объезд препятствий, смена полосы при соблюдении правил ПДД, является ключом к автономному управлению транспортными средствами и реализуется с помощью алгоритмов планирования и генерации траектории движения, включенных в программное обеспечение модуля, отвечающего за навигацию, интерпретацию ситуации и принятие решений автономного транспортного средства.

Планирование и генерация траектории движения - это процедура, требующая больших объемов памяти и существенных вычислительных ресурсов, которая должна выполняться на транспортном средстве в реальном режиме времени параллельно с другими вычислительными операциями, например, отслеживание препятствий, объединение данных с различных сенсоров и расчет модулей следования траектории. Входы и выходы процедуры планирования движения как правило зависят от данных генерируемых другими модулями: получают входные данные от модулей восприятия, обрабатывающих сырые данные с сенсоров, и дополняют их данными из цифровых дорожных карт, чтобы на основании всех потоков информации сформировать виртуальную картину мира, в котором необходимо проводить планирование [37].

Последовательность прямых и дуг с постоянным радиусом могут сгенерировать минимальные по длине траектории для маневров [38], но разрывы в кривизне могут привести к скачкообразному разрывному изменению угла

поворота руля, что потребует (теоретически) бесконечно большого управляющего воздействия. Также это приведет к тому, что БТС не сможет следовать по таким траекториям [39]. Отсюда следует, что для практического применения необходимо использовать кривые с непрерывной кривизной. Для этого часто используют клотоидные кривые [40], основанные на траекториях Дубина [41—43]. В работе [44] используются аналитические сплайны и эвристики для генерации гладкой траектории.

Маневрирование автономными системами в условиях препятствий является сложной задачей, которая возникает в различных практических приложениях, таких как роботы-манипуляторы и автономные системы, такие как самоуправляемые автомобили. Фундаментальной особенностью почти всех этих приложений является способность системы избегать столкновений с препятствиями. Такими препятствиями могут быть люди, автономные системы или статичные объекты, такие, как стены. Основная цель планирования траектории автономного транспортного средства - обеспечить его безопасное и беспрепятственное движение к месту назначения. Необходимо учитывать несколько важных факторов. Прежде всего, необходимо соблюдать правила дорожного движения, такие как ограничения скорости и сигналы светофора на перекрестках. Кроме того, решающее значение имеют физические характеристики транспортного средства, включая максимальную скорость, возможности ускорения, минимальный радиус поворота и маневренность в ограниченном пространстве.

Входные и выходные данные процесса планирования движения транспортного средства обычно зависят от данных, генерируемых другими компонентами системы. Эти компоненты предоставляют важную информацию, необходимую для правильного планирования маршрута и выполнения задач в режиме реального времени. Собранные данные поступают от компонентов восприятия, которые обрабатывают необработанные данные датчиков и объединяют их с информацией из цифровых карт для создания виртуального представления окружающей среды на основе всей доступной информации, которая затем используется для целей планирования.

Выпуклая оптимизация, известная своей способностью находить глобально оптимальные решения при относительно низких вычислительных затратах, хорошо подходит для задач планирования траектории движения

транспортных средств [45]. Однако для обеспечения безопасной и выполнимой траектории движения транспортного средства крайне важно учитывать границы дороги, которые формируют как геометрические, так и динамические ограничения [29]. Интеграция данных ограничений в постановку задачи выпуклой оптимизации позволяет существенно повысить способность транспортного средства формировать безопасные, гладкие и динамически реализуемые траектории в режиме реального времени [28].

Во многих традиционных подходах к планированию маршрутов задача формулируется в декартовой системе координат, в которой траектория движения транспортного средства описывается непосредственно в глобальном пространстве [15]. Несмотря на интуитивную наглядность такого представления, оно приводит к усложнению оптимизационных постановок, особенно при движении по криволинейным дорогам и при наличии пространственных ограничений, таких как границы полос движения и статические или динамические препятствия. В качестве альтернативы широко применяется криволинейная система координат, связанная с опорной линией дороги (система координат Френе), которая позволяет существенно упростить задачу планирования движения за счет согласования траектории транспортного средства с геометрией дороги [29]. Благодаря разделению движения автомобиля на продольную и поперечную составляющие относительно опорной траектории, использование системы координат Френе обеспечивает более компактную и вычислительно эффективную постановку задачи оптимизации, а также позволяет естественным образом формулировать ограничения, связанные с геометрией дороги и габаритами транспортного средства [28].

Квадратичное программирование (QP) является естественным выбором для постановки задачи планирования пути в системе координат, выровненной с дорогой (системе координат Френе), при использовании кинематической модели велосипеда. В этой постановке задачи мы формулируем квадратичную функцию цели (например, минимизация отклонения траектории, изменения кривизны или усилий управления) с линейными ограничениями (такими как ограничения на кривизну, скорость изменения кривизны, избегание препятствий и учет размеров автомобиля). Так как системы координат Френе позволяет выражать движение относительно опорного пути (обычно – центральной линии дороги), многие нелинейные динамические уравнения и геометрические ограничения

можно аппроксимировать линейными разложениями вокруг опорного пути. Результатом такой аппроксимации становится выпуклая оптимизационная задача, решаемая методом QR, что позволяет получить глобально оптимальное решение при высокой вычислительной эффективности. Как отмечается в ряде исследований, формулировка задачи планирования траектории в виде выпуклой QR позволяет обходиться без «раздувания» препятствий дополнительными запасами безопасности, что дает возможность максимально использовать доступное пространство даже в условиях значительных ограничений [46].

Задача настоящей главы состоит в сравнительном анализе существующих методов генерации траектории и в обосновании выбора подхода, используемого в диссертации для дальнейшей разработки. Для практического применения в составе АСУД требуется метод, который одновременно обеспечивает: явный учет дорожных границ и препятствий, формирование гладкой и динамически реализуемой траектории, а также предсказуемое время вычисления на бортовой аппаратуре.

Рабочая гипотеза главы заключается в том, что наиболее сбалансированным с инженерной точки зрения является двухэтапный подход: геометрическое формирование допустимого коридора движения и последующая выпуклая оптимизация траектории в координатах Френе. Такой подход выбран как основа дальнейшего исследования, поскольку он лучше согласуется с требованиями реального автомобиля по воспроизводимости и времени расчета, чем полностью дискретные или полностью нелинейные постановки.

Выбор данной тематики обусловлен прикладной необходимостью разработки алгоритма, который не только показывает результат в моделировании, но и может быть интегрирован в контур управления автомобиля с ограниченными вычислительными ресурсами и строгими требованиями к безопасности.

1.1 Базовые задачи движения беспилотного автомобиля

Беспилотный автомобиль представляет собой киберфизическую систему, обеспечивающую автономное движение в условиях динамически изменяющейся дорожной обстановки. В составе автоматизированной системы управления движением (АСУД) можно выделить взаимосвязанные функциональные подсистемы: получение и предварительную обработку данных сенсорного

комплекса, локализацию и построение представления окружающей среды, планирование движения, а также формирование управляющих воздействий для реализации выбранной траектории. Указанная функциональная структура обобщенно представлена на рисунке 1.

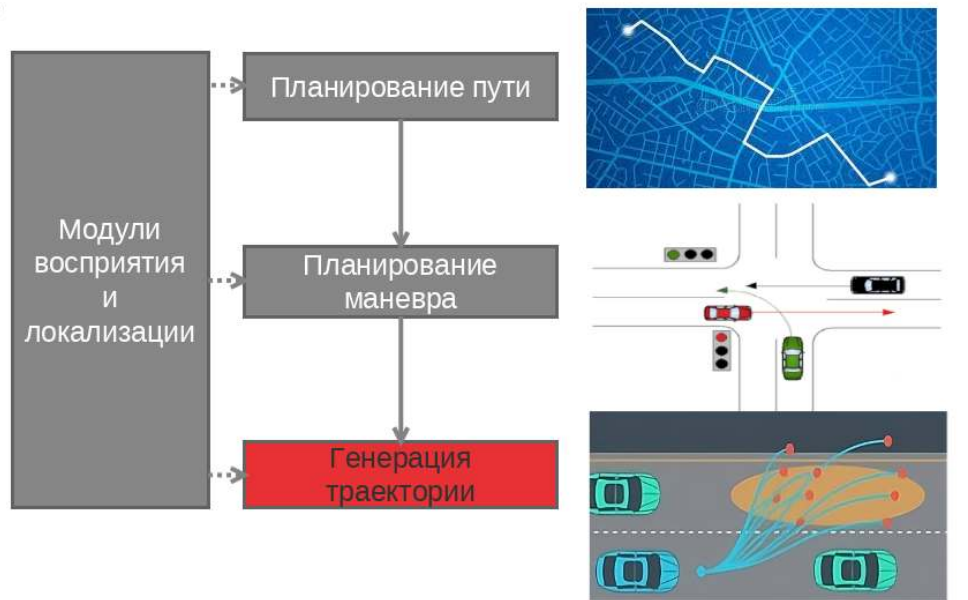


Рисунок 1 – Общая схема управления движением беспилотным ТС

С инженерной точки зрения ключевым элементом, определяющим безопасность и качество движения, является этап локального планирования и генерации траектории. На данном этапе по текущему состоянию транспортного средства (положение, скорость, ориентация), информации о конфигурации проезжей части (границы полосы/дороги, опорная линия) и данным о препятствиях формируется траектория движения в ближайшей окрестности автомобиля. Далее подсистема управления обеспечивает следование по сформированной траектории с учетом физико-динамических свойств транспортного средства.

Для бортовой реализации алгоритмов генерации траектории движения требуется одновременное выполнение следующих условий:

- геометрическая допустимость: нахождение траектории в пределах разрешенной области движения (границы полосы и дороги), с учетом габаритов транспортного средства;
- безопасность: отсутствие пересечений с областями, занятыми статическими препятствиями (и/или их безопасными аппроксимациями);

- кинематическая и динамическая реализуемость: соблюдение ограничений на кривизну траектории и скорость изменения кривизны, обусловленных возможностями рулевого управления и требованиями к поперечным ускорениям/комфорту движения;
- вычислительная эффективность: обеспечение расчета траектории в условиях ограниченных вычислительных возможностей и требований работы в реальном времени.

Анализ существующих подходов к планированию показывает, что существенная часть методов либо не обеспечивает удобного включения перечисленных ограничений в единую строгую постановку, либо обладает недостаточной предсказуемостью вычислительных затрат при работе в реальном времени [28]. В связи с этим перспективным направлением является применение оптимизационных методов, позволяющих формализовать критерии качества и ограничения движения, в том числе в системе координат, выровненной с дорогой, что упрощает учет дорожных границ и геометрии транспортного средства. Настоящая работа сосредоточена на этапе генерации траектории; в последующих разделах приведен обзор основных методов планирования и обоснована целесообразность оптимизационных постановок на основе квадратичного программирования.

1.2 Планирование траектории

Планирование движения беспилотного автомобиля целесообразно рассматривать как совокупность задач различного уровня детализации. На уровне маршрута формируется последовательность участков дорожной сети (глобальный путь), тогда как на локальном уровне строится траектория движения в ближайшей окрестности транспортного средства с высоким пространственным разрешением. В дальнейшем под планированием траектории понимается локальное построение движения на ограниченном горизонте, опирающееся на представление окружающей среды и проезжей части.

С практической точки зрения выбор алгоритма локального планирования определяется тем, в каком виде задана среда (точечные препятствия, карта занятости, поле стоимости, пространство конфигураций) и какой тип результата формируется на выходе (геометрическая кривая, последовательность путевых

точек, граф/дерево достижимости, коридор проходимости). В связи с этим методы планирования удобно разделять на следующие группы:

- методы, основанные на геометрических построениях и декомпозиции пространства (в частности, диаграммы Вороного), формирующие путь с максимизацией зазоров до препятствий;
- методы на основе потенциальных полей и полей стоимости, задающие направление движения через градиент “притяжения–отталкивания” и используемые для построения эвристик обхода препятствий;
- методы на основе дискретизации пространства (сетка занятости) с последующим поиском по графу/решетке, формирующие дискретный путь как последовательность узлов/ячеек свободного пространства;
- методы на основе выборки (RRT и модификации), строящие дерево достижимости в пространстве конфигураций и позволяющие получать допустимые решения в сложных пространствах.

Следует отметить, что перечисленные подходы часто формируют грубое геометрическое решение (путь/набор точек/коридор), пригодное как опорное для дальнейшей обработки. Получение же траектории, удовлетворяющей требованиям к гладкости и управляемости, как правило, выполняется отдельным этапом сглаживания и оптимизации. По этой причине далее (пп. 1.2.1-1.2.4) рассмотрены типовые методы построения опорного решения, а в последующих разделах главы приведено обоснование оптимизационных постановок (в том числе в форме задач квадратичного программирования) как инструмента получения управляемой траектории в условиях ограниченных вычислительных возможностей.

1.2.1 Диаграмма Вороного

Существующие методы планирования используют комбинацию различных алгоритмов для построения безопасного маршрута. Одним из таких алгоритмов является диаграмма Вороного [26]. Диаграмма Вороного конечного множества точек S на плоскости представляет такое разбиение плоскости, при котором каждая область этого разбиения образует множество точек, более близких к одному из элементов множества S , чем к любому другому элементу. Разбиение

плоскости (рисунок 2) осуществляется отрезками - ребрами, которые находятся на максимально возможном удалении от ближайших элементов множества S . Если элементы множества S – это препятствия, то ребра – это наиболее удаленные от препятствий точки. На рисунке 2 приведен пример маршрута, построенного с использованием диаграммы Вороного. Он проходит вдоль ребер диаграммы Вороного, что обеспечивает максимальную дистанцию от точек построенного маршрута до препятствий в каждый момент времени.

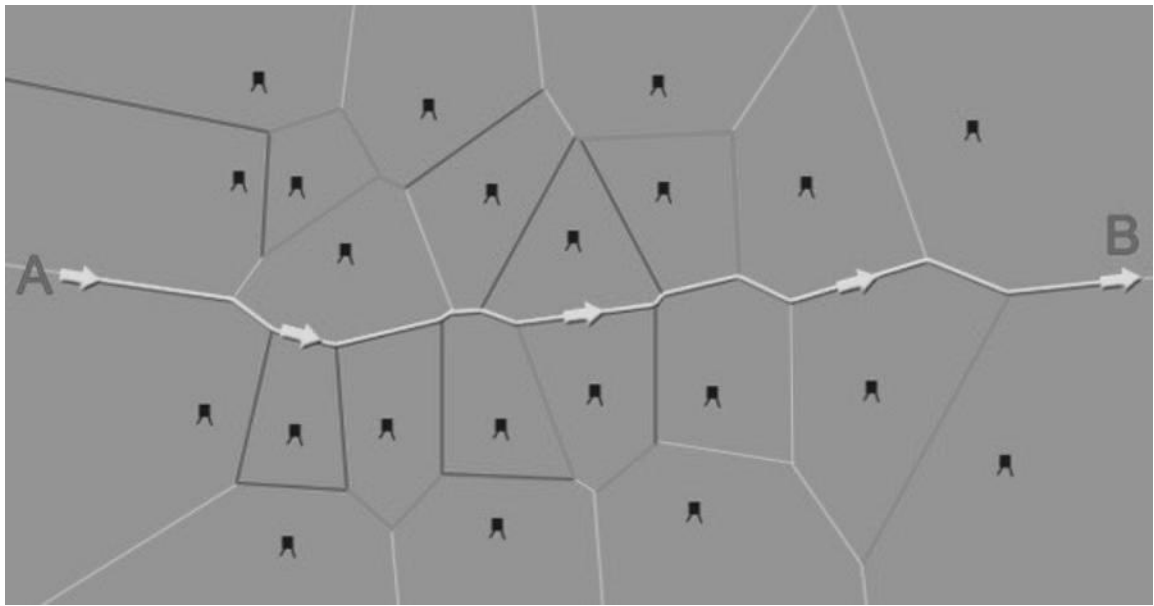


Рисунок 2 – Построенный маршрут с использованием диаграммы Вороного [47]

Исходя из рисунка 2 видно, как данная диаграмма позволяет найти путь, максимизирующий расстояние до препятствий. Однако, полученная траектория не обязательно проходит по проезжей части дороги или по одной ее полосе движения, в целом не гарантирована проходимость данной траектории автомобилем, и она не является оптимальной, так как на ней присутствуют резкие изломы. Именно поэтому диаграмма Вороного используется в качестве вспомогательной структуры данных, например, при комбинировании ее с методом потенциалов для получения поля Вороного, позволяющего лучше генерировать траектории движения для обхода препятствий.

На основании рисунка 2 можно заключить, что приведенная диаграмма предоставляет возможность найти траекторию, максимизирующую расстояние до окружающих препятствий. Однако такая траектория не обязательно находится в пределах проезжей части или одной полосы движения, что

затрудняет ее практическое использование. Более того, полученная траектория может содержать резкие угловые изломы, что снижает ее проходимость и маневренность для автомобиля. В связи с этим диаграмма Вороного применяется в качестве вспомогательной структуры данных. Например, ее часто комбинируют с методом искусственных потенциалов, что позволяет формировать так называемое поле Вороного [48]. Это поле обеспечивает более эффективную генерацию траекторий для обхода препятствий и улучшает плавность маршрутов, учитывая реальные условия движения.

Следует отметить, что диаграмма Вороного является геометрическим инструментом поиска пути с максимальным зазором до препятствий, однако она не обеспечивает выполнения ограничений, критичных для беспилотного автомобиля: соблюдения границ полосы движения, ограничений на кривизну и ее изменение, а также детерминированного качества траектории без процедуры сглаживания. По этой причине на практике диаграмма Вороного используется преимущественно как вспомогательная структура (например, для построения эвристик или полей стоимости), тогда как формирование управляемой траектории выполняется отдельным этапом оптимизации.

1.2.2 Метод потенциалов

Данный метод, описанный в работе А.К. Платонова, А.А. Кирильченко и М.А. Колганова [10], относится к классу «методов потенциалов» для решения задачи выбора пути мобильного робота. Суть метода заключается в моделировании движения робота в виртуальном поле «информационных сил», где целевая точка создает эффект притяжения, а препятствия — эффект отталкивания. Такой подход позволяет формализовать навигацию робота как движение подвижной точки, которая стремится к цели, одновременно избегая столкновений.

Основная идея состоит в том, что цель и препятствия рассматриваются как источники электрических зарядов: цель — с положительным зарядом, препятствия — с отрицательным. Подвижная точка (робот) получает отрицательный заряд и помещается в исходную позицию. В результате взаимодействия сил притяжения и отталкивания эта точка движется, комбинируя влияние «притягивающих» и «отталкивающих» сил. Законы движения подвижной точки могут задаваться по-разному, что позволяет варьировать

траектории с различной степенью близости к препятствиям — от более безопасных до более рискованных обходов. При определенных условиях и ограничениях на структуру местности и правила движения, подвижная точка сможет достигнуть цели.

В работе сравниваются различные способы построения потенциальных полей, например, с использованием диаграмм Вороного, которые формируют потенциальное поле для навигации. На рис. 3 (а) показано поле Вороного, являющееся потенциальным полем, построенным с помощью диаграммы Вороного, на рис. 3 (б) — диаграмма Вороного, на рис. 3 (в) — потенциальное поле. На рис. 3 представлено сравнение данных полей.

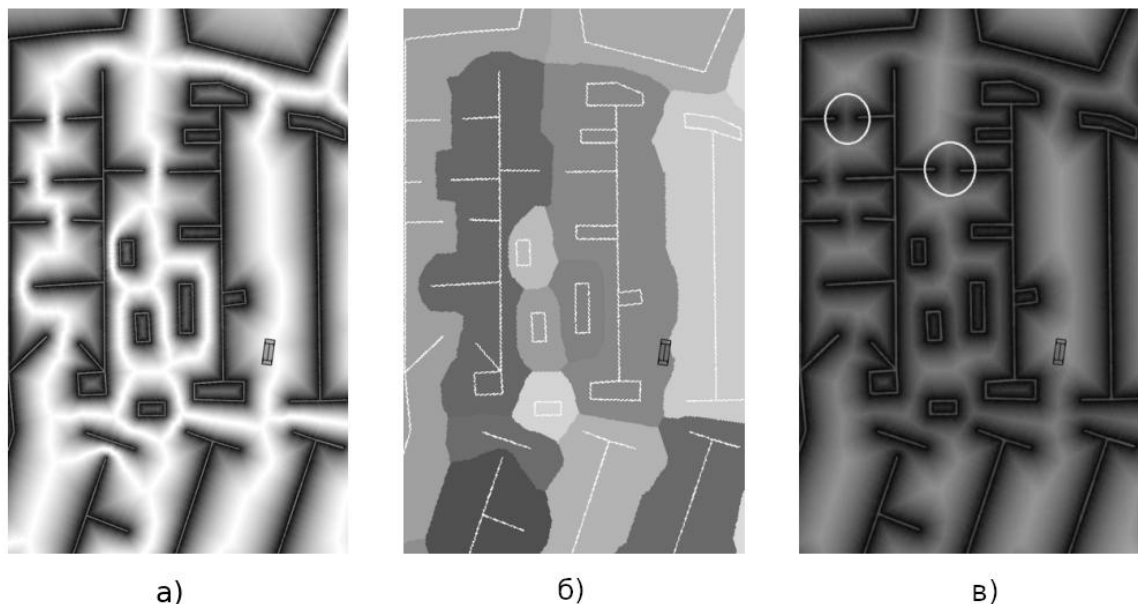


Рисунок 3 – (а) Поле Вороного, построенное для парковки. (б) Соответствующая диаграмма Вороного, по которой строилось поле. (в) Потенциальное поле [48]

Рисунок 3 (в) показывает потенциальное поле, полученное стандартным путем, имеет затемнения в узких проходах (помечено кругами), что может помешать алгоритму планирования движения проложить путь через этот проход, хотя физически автомобиль может быть способен там пройти. С полем Вороного такой проблемы не возникает (см. рисунок 3 (а) в том же узком месте).

С помощью поля Вороного возможно обеспечить оптимально безопасное перемещение автомобиля относительно других участников движения.

Ограничением методов потенциальных полей является возможность возникновения локальных минимумов и высокая чувствительность к выбору

параметров потенциалов, что затрудняет получение устойчивых траекторий в дорожных сценариях с узкими проездами и сложной геометрией. Кроме того, включение требований к движению в пределах полосы и ограничений на кривизну траектории в рамках потенциалов, как правило, носит эвристический характер. Поэтому методы потенциалов целесообразно рассматривать как средство построения поля стоимости/направления, тогда как итоговая траектория должна формироваться с использованием постановки, допускающей явный учет ограничений и критериев гладкости.

1.2.3 Сетка занятости

Альтернативным методом планирования маршрута автономного транспортного средства является использование сетки занятости [49, 50]. Данный метод позволяет дискретизировать пространство в сетку. Каждой ячейке данной сетки присваивается значение, определяющее степень занятости (наличия препятствий) в данной ячейке. Значения могут быть как дискретны: 0 – препятствие отсутствует, 100 – ячейка занята препятствием, так и носить вероятностный характер того, то ячейка занята, то есть хранить в себе значение вероятности от 0.0 до 1.0, где 0.0 – ячейка свободна, 1.0 – ячейка точно занята, 0.5 – состояние ячейки не известно (может быть и занята, и свободна). На рисунке 4 приведен пример представления мира в виде сетки занятости.

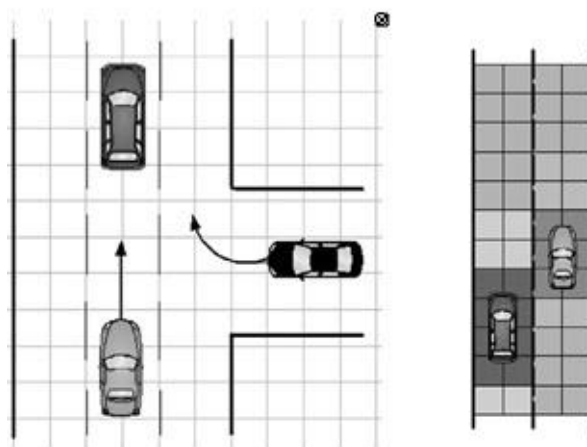


Рисунок 4 – Представление мира в виде сетки занятости. Слева – сетка, наложенная на дорожное пространство. Справа – раскрашенная сетка; цвета отражают степень занятости клетки [51]

В представлениях в виде сетки окружающая среда задается дискретно, и планирование пути сводится к поиску последовательности ячеек (или узлов графа), принадлежащих свободному пространству, с минимизацией выбранного функционала стоимости. Достоинством подхода является удобство строгого учета препятствий и возможность построения коридора проходимости на основе карты занятости или производных от нее карт (например, карт расстояний/стоимости). Вместе с тем дискретная природа решения приводит к ломаному характеру пути, что требует последующего сглаживания и приведения траектории к виду, удовлетворяющему ограничениям на кривизну и ее изменение. Сетка занятости целесообразна как источник геометрических ограничений и области допустимого движения, тогда как формирование гладкой траектории следует выполнять оптимизационными методами с явным учетом указанных ограничений.

1.2.4 Метод случайных деревьев

При использовании методов на основе быстро исследующих случайных деревьев (rapidly-exploring random trees, RRT) строится древовидная структура в пространстве конфигураций/состояний, которая итеративно расширяется за счет добавления случайно сэмплированных конфигураций из области свободного пространства [25]. Базовый алгоритм RRT (рисунок 5) ориентирован на нахождение допустимой траектории и обладает свойством вероятностной полноты при выполнении определенных условий, однако не гарантирует оптимальности; асимптотическая оптимальность достигается в модификациях типа RRT*.

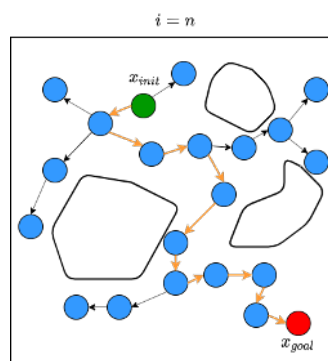


Рисунок 5 – Схема работы RRT [52]

Данный метод является вероятностным и может быть реализован в процессе движения в реальном времени, несмотря на то что он работает с динамической системой, у которой в каждый момент времени меняются входные данные. Однако недостатком данного метода является необходимость проверки коллизий для каждого расширенного узла. Коллизии возникают при увеличении препятствий и участников движения. В данном контексте под коллизией понимается проблема обнаружения пересечений между собой двух или более путей.

Ключевым ограничением методов случайных деревьев для задачи движения автомобиля является слабый контроль гладкости траектории и отсутствие детерминированного времени вычисления: качество решения и затраты времени зависят от числа выборок и эвристик расширения дерева. Учет кинематических и динамических ограничений (минимальный радиус поворота, ограничения на кривизну/ускорения) требует усложнения процедуры управления рулем, что повышает вычислительные затраты. По этой причине траектории, полученные методами RRT и их модификациями, в практических системах часто используются как предварительное решение, требующее последующей оптимизации и сглаживания с явным учетом дорожных ограничений и ограничений управляемости.

1.3 Локализация и картография

Автономное движение беспилотного автомобиля требует получения достоверной оценки текущего состояния транспортного средства и формирования представления о дорожной среде, достаточного для последующего планирования траектории. В контуре автономного управления результаты подсистем локализации и картографии используются как исходные данные для планировщика: оценка позы (положения и ориентации) и скорости автомобиля задает начальные условия, а представление дороги и препятствий определяет область допустимого движения и ограничения, накладываемые на формируемую траекторию.

Под локализацией далее понимается задача оценки положения и ориентации автомобиля в пространстве и (при необходимости) параметров движения. На практике применяются:

- глобальная локализация (спутниковые навигационные системы и

инерциальные измерительные устройства), обеспечивающая привязку к глобальной системе координат; однако она подвержена деградации точности в условиях плотной городской застройки, тоннелей и многолучевости;

- локальная локализация (визуальная одометрия, SLAM), ориентированная на уточнение позы в сложных условиях за счет сопоставления наблюдений с локальной/глобальной картой; при этом SLAM позволяет одновременно уточнять положение и строить карту окружающей среды;

- сенсорная фильтрация и объединение данных (например, фильтр Калмана и частицный фильтр), повышающие точность и устойчивость оценки за счет интеграции информации от различных сенсоров.

Под картографическим обеспечением понимается формирование и использование представления дорожной среды, включающего (в зависимости от используемой архитектуры) параметры дорожной инфраструктуры (геометрия проезжей части, границы полос, опорные линии) и локальное представление препятствий (например, в виде карты занятости или близких по смыслу структур) [53]. Для задач генерации траектории принципиально важно, что именно картографическое представление задает геометрические ограничения движения (границы дороги/полос) и ограничения по препятствиям, которые в дальнейшем должны быть учтены при формировании управляемой траектории.

В рамках настоящей работы подсистемы локализации и картографии рассматриваются как источники входных данных, обеспечивающие оценку текущего состояния транспортного средства и описание области допустимого движения. Основное внимание далее уделяется методам построения траектории с явным учетом дорожных ограничений и препятствий, а также требованиям к гладкости и реализуемости траектории, что соответствует постановкам на основе оптимизации, рассматриваемым в последующих разделах.

1.4 Управление движением

Подсистема управления беспилотного автомобиля обеспечивает выполнение заданной (опорной) траектории, сформированной модулем планирования, с учетом динамических свойств транспортного средства, ограничений исполнительных механизмов и требований к устойчивости. Задача управления в типовой постановке сводится к минимизации ошибок слежения по

положению и ориентации относительно опорной траектории, а также к формированию управляющих воздействий (угол поворота управляемых колес, тяговое и тормозное воздействие), обеспечивающих движение по траектории на заданном горизонте.

С практической точки зрения выделяют продольное и поперечное управление. Продольное управление обеспечивает поддержание требуемой скорости/ускорения, поперечное — компенсацию отклонений от опорной траектории по боковому смещению и курсовому углу. Наиболее распространенные подходы к управлению траекторией включают:

- Геометрические регуляторы (например, Pure Pursuit, Stanley), которые формируют управляющее воздействие из геометрических ошибок относительно опорного пути [17, 54]. Эти методы отличаются простотой и низкими вычислительными затратами, однако чувствительны к выбору параметров и, как правило, не обеспечивают явного учета ограничений на управляющее воздействие и динамику.

- Линейно-квадратичные методы (LQR), основанные на линеаризации модели в окрестности траектории и минимизации квадратичного функционала ошибки и управления [55]. LQR обеспечивает вычислительно эффективный закон управления и широко используется как базовый метод или эталон для сравнения.

- Предиктивное управление (MPC), в котором на каждом шаге решается оптимизационная задача на конечном горизонте с учетом ограничений на состояние и управляющие воздействия. MPC позволяет явно учитывать насыщения и ограничения, обеспечивая повышенную точность слежения в условиях сложной траектории и возмущений [28].

Следует отметить, что эффективность подсистемы управления существенно зависит от свойств опорной траектории: наличие разрывов кривизны, резких изменений направления или недостаточный зазор до препятствий приводит к росту ошибок слежения и к возникновению режимов насыщения рулевого управления. Поэтому в практических системах автономного вождения требования к управляемости переносятся на этап планирования и сглаживания траектории, где обеспечивается формирование

гладкой, геометрически допустимой и динамически реализуемой траектории, пригодной для исполнения в контуре управления [28, 29].

В рамках настоящей работы детальный анализ и сравнение алгоритмов управления траекторией (включая LQR и MPC) приводится далее; при этом для экспериментальных исследований выбран подход на основе MPC как обеспечивающий более высокую точность следования опорной траектории при наличии ограничений.

1.5 Преимущества использования квадратичного программирования для реального времени

Задачи квадратичного программирования (QP) при выпуклой постановке допускают применение специализированных численных методов, обеспечивающих приемлемую вычислительную сложность для бортовых систем автономного управления. Практически важным является использование разреженной структуры матриц и повторяемости постановки на последовательных тактах планирования, что позволяет применять инициализацию предыдущим решением (warm-start) и тем самым снижать среднее время расчета при работе на фиксированном горизонте.

Для выпуклой QP обеспечивается глобальная оптимальность решения относительно заданного функционала и ограничений при наличии допустимой области. Это повышает надежность вычислительного процесса по сравнению с общими нелинейными постановками, где возможны локальные минимумы и существенная вариативность времени сходимости. В системах реального времени данное свойство является ключевым, поскольку позволяет обеспечить повторяемость поведения алгоритма на последовательных шагах планирования.

QP удобна для явного задания ограничений в виде линейных неравенств, что соответствует практической необходимости учитывать:

- границы полосы движения/дороги (в т.ч. через аппроксимацию допустимой области выпуклыми множителями);
- запретные области, связанные со статическими препятствиями (через построение ограничений по свободному пространству);
- ограничения управляемости, задаваемые допустимыми диапазонами кривизны и (при необходимости) скорости изменения кривизны.

Данная форма ограничений обеспечивает прямую интеграцию требований безопасности и реализуемости в одну оптимизационную постановку.

Квадратичная функция цели позволяет формализовать компромисс между несколькими критериями качества: отклонение от опорной линии/референса, сглаживание формы траектории (включая штрафы на кривизну и ее изменения), а также регуляризацию управляющих/производных величин. В отличие от методов, формирующих дискретный или ломаный путь (графовый поиск, сэмплирование), оптимизационная постановка позволяет получать траекторию как единое согласованное решение с заданными свойствами гладкости.

Таким образом, квадратичное программирование представляет вычислительно эффективный и формализуемый аппарат для построения управляемых траекторий в режиме реального времени при наличии ограничений дорожной среды. Далее приведется общая постановка задачи QP и обсуждаются особенности ее решения выбранным численным решателем.

1.6 Сравнение с другими подходами

Для оценки применимости методов планирования и формирования траектории в задачах автономного вождения целесообразно использовать следующие критерии: предсказуемость вычислительных затрат в реальном времени, возможность явного учета ограничений дорожной инфраструктуры и препятствий, обеспечение гладкости и управляемости формируемой траектории, устойчивость к изменениям сцены и наличие гарантий по качеству решения.

В сравнении с дискретными методами (графовый поиск, решеточные/lattice-планировщики) и методами сэмплирования (RRT и др.), указанные подходы эффективно решают задачу поиска допустимого пути в сложных сценариях и удобны для построения коридора проходимости, однако получаемое решение нередко имеет дискретный/ломаный характер и требует проведения отдельного этапа сглаживания. Кроме того, вычислительное время и качество решения могут существенно зависеть от разрешения дискретизации и эвристик поиска/сэмплирования. В QP-подходе ограничения и критерии гладкости задаются в рамках единой постановки, что упрощает получение траектории, пригодной для исполнения в контуре управления.

Нелинейные постановки оптимального управления позволяют точнее учитывать динамику и нелинейные ограничения, однако на практике требуют

больших вычислительных затрат и могут демонстрировать чувствительность к начальному приближению, а также вариативность времени сходимости. Для задач реального времени целесообразен компромиссный подход, при котором используется выпуклая аппроксимация и решается QP-задача, обеспечивающая приемлемую вычислительную сложность и предсказуемость вычислений.

Реактивные методы и системы на основе правил обладают высокой скоростью принятия решений, но, как правило, не обеспечивают оптимальность и формализуемый баланс между критериями качества (плавность, комфорт, отклонение от референса) при одновременном строгом учете ограничений. Это может приводить как к излишне консервативному поведению, так и к ухудшению качества формирования траектории в условиях плотного трафика. Оптимизационный QP-подход позволяет точно задавать критерии и ограничения, обеспечивая воспроизводимое качество траектории в типовых сценариях движения.

Следует отметить, что в практических системах планирования широко применяется двухэтапная схема: поиск пути (геометрического решения без столкновений) и последующая генерация/сглаживание траектории. Алгоритмы поиска (графовые, на основе выборки и др.) могут эффективно находить путь без столкновений, однако такие решения часто требуют последующей обработки для обеспечения непрерывности и ограничений управляемости. Параметрические кривые (полиномы, сплайны, кривые Безье) обеспечивают непрерывность траектории, но без дополнительных процедур не гарантируют удовлетворения ограничениям кинематической и динамической реализуемости. В этом контексте методы выпуклой оптимизации (QP) целесообразно рассматривать как вычислительно реализуемую основу для этапа формирования и сглаживания траектории при наличии ограничений дорожной среды. В таблице 1 представлена сводка по классам методов формирования траектории.

Таблица 1 – Сравнение классов методов формирования траектории

Класс методов	Учет ограничений	Гладкость траектории	Предсказуемость времени
Графовые/решеточные	Высокий (дискретно), но требует постобработки	Низкая без сглаживания	Зависит от разрешения и эвристик
Методы на основе выборки (RRT)	Умеренный; зависит от процедуры локального соединения и проверки на столкновения	Низкая без постобработки	Варьируется; не детерминировано
NLP-оптимизация	Высокий (в том числе нелинейные ограничения)	Высокая	Часто высокая вычислительная стоимость; вариативная сходимость
QP (выпуклая)	Высокий для линейных/выпуклых аппроксимаций	Высокая (за счет квадратичных штрафов)	Высокая; пригодна для задач реального времени

1.6.1 Ограничения существующих и предлагаемого подходов при генерации локальной траектории движения автомобиля

Сводный анализ показывает, что на этапе внедрения алгоритмов в бортовую систему управления ключевые ограничения связаны не только с качеством траектории, но и с вычислительной стабильностью и интерпретируемостью ограничений.

- **Графовые/решеточные методы:** удобны для поиска допустимого пути, но чувствительны к шагу дискретизации и обычно требуют отдельного этапа сглаживания для соблюдения ограничений по кривизне и управляемости.

- **Sampling-based методы:** хорошо работают в сложной геометрии, однако дают вариативное время расчета и затрудняют прямое задание ограничений на динамические параметры движения.

- **Нелинейная оптимизация общего вида:** обеспечивает высокий уровень детализации модели, но может быть вычислительно тяжелой и чувствительной к начальному приближению, что усложняет применение в жестком цикле реального времени.

• **Предлагаемый QR-подход в координатах Френе:** обеспечивает формализуемый учет ограничений и предсказуемое время решения, но требует корректной локализации, достоверной геометрии дорожного коридора и соблюдения области применимости линеаризованной модели (без режимов интенсивного бокового скольжения и срыва сцепления).

Следовательно, предлагаемый подход не устраняет полностью инженерные риски, но обеспечивает более управляемый компромисс между качеством траектории, безопасностью и вычислительной реализуемостью на целевой аппаратуре.

Таким образом, применение выпуклой квадратичной оптимизации целесообразно рассматривать как вычислительно реализуемую основу на этапе формирования и сглаживания траектории при наличии ограничений дорожной среды. На следующих этапах разработки траектория и ограничения будут приводиться к единой канонической постановке квадратичного программирования, в которой требования к качеству движения (гладкость, близость к референсу, ограничения по управлению) задаются квадратичной целевой функцией, а ограничения среды и реализуемости — системой линейных ограничений. Далее (п. 1.7) приведена общая математическая форма задачи QR, используемая в работе.

1.7 Формулировка задачи квадратичного программирования в общем виде

В данной работе задачи формирования/сглаживания траектории приводятся к выпуклой задаче квадратичного программирования (QR) в канонической форме, удобной для численного решения в реальном времени. В общем виде рассматривается задача

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & \frac{1}{2} x^T P x + q^T x, \\ \text{s.t.} \quad & l \leq A x \leq u, \end{aligned} \tag{1.1}$$

где $x \in \mathbb{R}^n$ — вектор оптимизируемых переменных; $P \in \mathbb{R}^{n \times n}$ — симметричная матрица квадратичных коэффициентов, $P = P^T$; $q \in \mathbb{R}^n$ — вектор линейных коэффициентов; $A \in \mathbb{R}^{m \times n}$ — матрица линейных ограничений; $l, u \in (\mathbb{R} \cup \{-\infty, +\infty\})^m$ — нижние и верхние границы на левую часть

ограничений.

Постановка (1.1) является выпуклой при условии $P \succeq 0$ (положительной полуопределенности), что исключает локальные минимумы, отличные от глобального. Равенства и неравенства задаются единообразно через границы: равенство соответствует $l_i = u_i$, одностороннее неравенство реализуется установкой одной из границ равной $\pm\infty$. Такая запись удобна для компактного представления ограничений дорожного коридора, предельных значений управляющих воздействий, а также линейных аппроксимаций нелинейных ограничений.

Для решения задач вида (1.1) в работе используется решатель OSQP (Operator Splitting Quadratic Program) [56]. Данный решатель ориентирован на большие разреженные QP-задачи с линейными ограничениями и допускает эффективное обновление параметров постановки (векторов q, l, u) при фиксированной структуре матриц P и A , что является принципиальным для применения в задачах реального времени.

Далее (п. 1.8) рассматривается параметризация траектории в координатах Френе, позволяющая естественным образом сформировать целевую функцию и ограничения задачи (1.1) с учетом геометрии дороги и границ проходимого коридора.

1.8 Использование координат Френе

Для формализации задачи планирования траектории в дорожной среде целесообразно использовать систему координат Френе, задаваемую относительно гладкой опорной (референсной) траектории

$$C : s \mapsto (x_r(s), y_r(s), \psi_r(s)),$$

параметризованной длиной дуги s . Такая параметризация позволяет описывать положение транспортного средства не в абсолютной декартовой системе (x, y) , а в координатах дороги, что упрощает постановку ограничений и критериев качества.

В системе Френе состояние представляется через продольную координату s - пройденное расстояние вдоль опорной траектории, и поперечное отклонение d — знаковое расстояние от опорной траектории по нормали (по принятому соглашению $d < 0$ соответствует смещению влево, $d > 0$ — вправо). Для

описания ориентации вводится угловая ошибка e_ϕ , определяемая как разность между текущим курсом автомобиля и направлением касательной к опорной траектории в точке проекции. Геометрия движения дополнительно характеризуется кривизной κ траектории объекта.

На рисунке 6 приведено сопоставление декартовой системы координат и системы координат Френе. В декартовых координатах движение вдоль криволинейной дороги требует совместного описания $x(t)$ и $y(t)$, тогда как в системе координат Френе динамика в окрестности опорной траектории сводится к управлению малыми отклонениями $d(t)$ и $e_\phi(t)$.

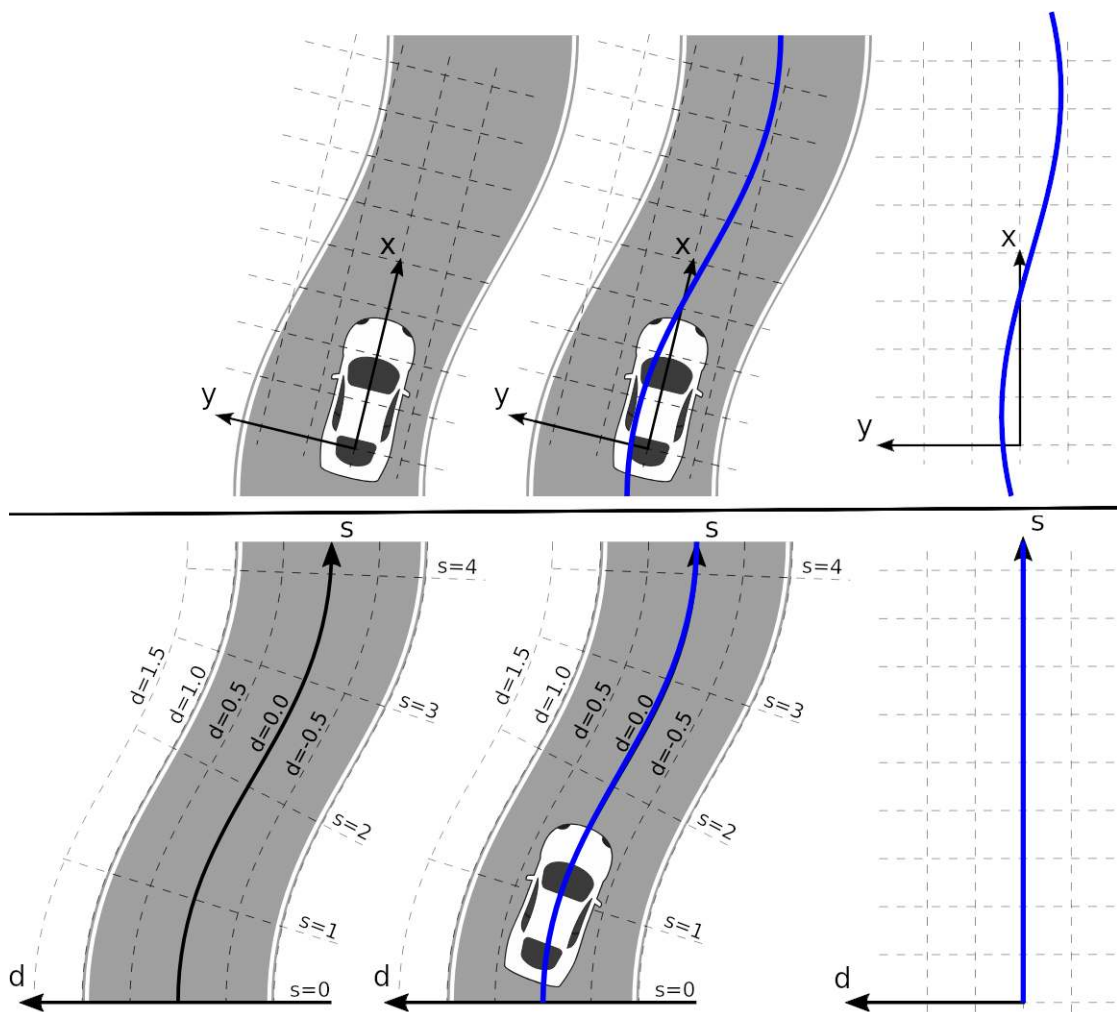


Рисунок 6 – Сравнение декартова пространства и системы отсчета Френе [15]

Связь между декартовыми координатами точки (x, y) и параметризацией Френе (s, d, e_ϕ, κ) задается через геометрию опорной траектории:

$$x = x_r(s) - d \sin \psi_r(s), \quad (1.2)$$

$$y = y_r(s) + d \cos \psi_r(s), \quad (1.3)$$

$$\psi = \psi_r(s) + e_\phi, \quad (1.4)$$

$$\kappa = \frac{\kappa_r(s) + \dot{e}_\phi / \dot{s}}{1 - \kappa_r(s) d}, \quad (1.5)$$

где $\kappa_r(s)$ — кривизна опорной траектории, $(\dot{\cdot})$ — производная по времени. Из (1.5) следует, что корректность преобразования требует выполнения условия $|\kappa_r(s) d| < 1$, которое на практике обеспечивается движением в пределах дорожного коридора (в окрестности осевой линии).

Важное преимущество фрейма Френе для задач автономного вождения состоит в том, что ограничения дорожной инфраструктуры естественным образом выражаются как ограничения на поперечное положение:

$$d_{\min}(s) \leq d(s) \leq d_{\max}(s),$$

а требования к плавности и управляемости траектории — как штрафы и/или ограничения на d , e_ϕ и κ . Тем самым ввод координат Френе обеспечивает переход от геометрически сложных ограничений в (x, y) к простым линейным/квазилинейным условиям в «координатах дороги», что непосредственно согласуется с постановкой задач в виде квадратичного программирования, введенной в п. 1.7.

Параметризация в координатах Френе широко применяется в задачах планирования траектории и управления автономными транспортными средствами, поскольку обеспечивает естественную формализацию ограничений и критериев оптимальности в системе координат дороги [28, 29]. Далее на этой основе вводится используемая в работе модель движения в фрейме Френе и приводится ее форма, пригодная для применения в алгоритмах генерации траектории.

1.9 Выводы

В результате выполнения обзорного анализа в главе 1 сделаны следующие выводы.

1. Выполнен анализ отечественных и зарубежных исследований, посвященных методам локального планирования и генерации траектории движения высокоавтоматизированных и беспилотных транспортных средств в условиях городской дорожной среды, а также рассмотрены функциональные задачи, определяющие требования к формированию траектории при наличии ограничений дорожной инфраструктуры и окружающей обстановки. Показано, что практическая значимость рассматриваемой задачи определяется высокой долей ДТП, связанных с человеческим фактором: около 94% серьезных ДТП в США и около 88% ДТП в Российской Федерации обусловлены ошибками водителей. Дополнительно установлено, что в условиях, характерных для эксплуатации в Российской Федерации, фактическая эффективность серийных систем активной безопасности составляет лишь около 57% предотвращенных типовых столкновений при потенциально достижимом уровне порядка 97%, что указывает на наличие существенного разрыва между требуемой и достигаемой эффективностью.
2. Проведена классификация основных подходов к планированию траектории (дискретно-графовые методы, методы потенциальных полей, вероятностные методы выборки, подходы оптимального управления и оптимизационные постановки). Показано, что при практическом применении существенная часть известных решений испытывает ограничения при одновременном обеспечении требований реального времени, явного учета границ полосы движения и препятствий, а также требований к гладкости траектории (включая ограничения по кривизне и скорости изменения кривизны). Для графовых и решеточных методов конкретными недостатками являются зависимость качества решения от шага дискретизации, кусочно-линейный характер траектории и необходимость последующего сглаживания; для полиномиальных и сплайновых параметризаций — косвенный контроль кривизны и расстояния до границ полосы, что может приводить к локальным

превышениям ограничений.

3. Установлено, что методы дискретного поиска и вероятностной выборки, как правило, характеризуются вариативностью вычислительных затрат и требуют дополнительных процедур сглаживания и обеспечения динамической реализуемости траектории. Методы потенциальных полей обладают ограниченной устойчивостью к локальным минимумам и чувствительны к параметрам настройки. Подходы на основе сеточных представлений и карт занятости удобны для описания сцены, однако также требуют последующего построения гладкой и управляемой траектории с учетом дорожных границ и свободного пространства. Для sampling-based методов конкретными недостатками являются недетерминированное время расчета и сложность прямого учета динамических ограничений; для общих нелинейных оптимизационных постановок — высокая вычислительная стоимость и чувствительность к начальному приближению; для методов, основанных на обучении, — необходимость значительных объемов обучающих данных, сложность формальной верификации и отсутствие строгих гарантий по кривизне, комфортности и соблюдению габаритных ограничений в общем виде.
4. Показано, что постановки, сводимые к задачам выпуклой оптимизации, в частности квадратичного программирования, являются перспективной основой для алгоритмов локального планирования, поскольку позволяют в явном виде учитывать ограничения, накладываемые геометрией дороги, границами полос движения и препятствиями, и, одновременно, формировать траекторию с заданными требованиями по гладкости при приемлемой вычислительной сложности и предсказуемости времени расчета.
5. Отмечено, что использование координат дороги (в том числе координат Френе) обеспечивает удобную параметризацию движения относительно опорной траектории и позволяет естественным образом задавать дорожные ограничения и критерии качества в переменных, обладающих прямой геометрической интерпретацией. Тем самым, результаты проведенного анализа позволяют обосновать выбор оптимизационного аппарата выпуклой оптимизации и переход к

построению математических моделей и алгоритмов следования траектории движения автомобиля.

Научная новизна предлагаемого подхода состоит не в самостоятельном применении системы координат Френе или методов квадратичного программирования, а в разработанном способе сопряжения геометрического и оптимизационного этапов локального планирования. На первом этапе по данным о границах проезжей части и статических препятствиях формируется связный допустимый коридор движения с учетом габаритов автомобиля. На втором этапе границы полученного коридора представляются в виде линейных ограничений выпуклой задачи квадратичного программирования, в которой синтезируется траектория с ограничениями на кривизну и скорость ее изменения. Применение модифицированного функционала качества и фиксированного горизонта планирования обеспечивает одновременное соблюдение требований геометрической безопасности, динамической реализуемости и вычислительной эффективности в режиме реального времени.

1.10 Цель и задачи исследования

Проведенный в главе 1 анализ отечественных и зарубежных подходов к локальному планированию и формированию траектории показал, что существенная часть решений либо затрудняет совместный учет ограничений дорожной инфраструктуры и геометрии транспортного средства в рамках единой строгой постановки, либо характеризуется недостаточной предсказуемостью вычислительных затрат в режиме реального времени. С учетом изложенного, а также с учетом целесообразности использования координат Френе для явного задания дорожных границ и геометрически интерпретируемых ограничений, формулируются цель и задачи диссертационной работы.

Цель исследования сформулирована во Введении и заключается в повышении безопасности и качества автоматического движения автомобиля за счет разработки метода и реализующего его алгоритма синтеза локальной траектории движения автомобиля на основе выпуклого квадратичного программирования, обеспечивающих формирование в режиме реального времени гладкой и геометрически допустимой траектории с учетом границ проезжей части, статических препятствий, габаритов автомобиля, ограничений

на кривизну и ее производную по дуговой координате.

Предлагаемый алгоритм направлен на обеспечение движения в пределах заданной полосы с возможностью маневра безопасного перестроения, на снижение изменений кривизны траектории и поддержание комфортных динамических параметров движения БТС. Построение траектории движения осуществляется на основе параметризованной кинематической модели автомобиля и формулируется как задача выпуклой оптимизации, допускающей эффективное численное решение в рамках ограниченного вычислительного бюджета, соответствующего требованиям к бортовым системам управления. Разрабатываемое решение должно обеспечивать адаптацию движения БТС к криволинейной геометрии дороги, учитывать габариты автомобиля без избыточного увеличения вычислительной нагрузки системы управления движением и демонстрировать устойчивость и применимость при эксплуатации в условиях городской дорожной среды.

Для достижения поставленной цели необходимо решить следующие **задачи**:

1. Провести анализ и классификацию методов и алгоритмов локального планирования и управления траекторным движением высокоавтоматизированных и беспилотных автомобилей; установить ограничения известных подходов при учете геометрии проезжей части, габаритов автомобиля, статических препятствий и динамических ограничений; сформулировать требования к разрабатываемому методу.

2. Обосновать выбор и выполнить адаптацию математических моделей автоматизированного управления движением автомобиля для решения задач локального планирования и траекторного управления в декартовой системе координат и системе координат Френе; сформировать систему геометрических, кинематических и динамических ограничений.

3. Разработать метод и алгоритм синтеза закона траекторного управления автомобилем на основе оптимального управления; сформировать критерий качества, учитывающий точность следования заданной траектории, интенсивность и плавность рулевого воздействия, а также ограничения исполнительного механизма и поперечной динамики автомобиля; выполнить сравнительное исследование законов управления в единых условиях

моделирования.

4. Разработать метод и алгоритм синтеза локальной траектории движения автомобиля на основе выпуклой квадратичной оптимизации, включающий формирование допустимого коридора движения с учетом границ проезжей части, статических препятствий и габаритов автомобиля, а также ограничения на кривизну траектории и скорость ее изменения; исследовать влияние параметров функционала и горизонта планирования на качество и вычислительную реализуемость решения.

5. Реализовать разработанные методы и алгоритмы его применения в составе программно-алгоритмического комплекса автоматизированной системы управления движением и провести их имитационную и натурную верификацию на автомобиле «Лада Веста» по показателям точности следования траектории, геометрической и динамической реализуемости, безопасности объезда препятствий и времени вычисления.

ГЛАВА 2. Математическая модель автомобиля и алгоритмы следования траектории

Данная глава посвящена нижнему уровню контура управления движением: математической модели автомобиля, построению ошибок слежения и синтезу алгоритмов управления траекторией — алгоритма чистого преследования Pure Pursuit (PP), линейно-квадратичного регулятора (LQR) и модельно-предиктивного управления (Model Predictive Control, MPC). Входом поперечных регуляторов является опорная траектория, сформированная модулем планирования и содержащая геометрические параметры и заданный продольный профиль. Выходом поперечного регулятора является угол поворота управляемых колес δ ; слежение за заданной продольной скоростью выполняется отдельным контуром, формирующим воздействие на продольный привод.

Функциональная схема с указанием места рассматриваемой части в общей архитектуре АСУД показана на рисунке 7.

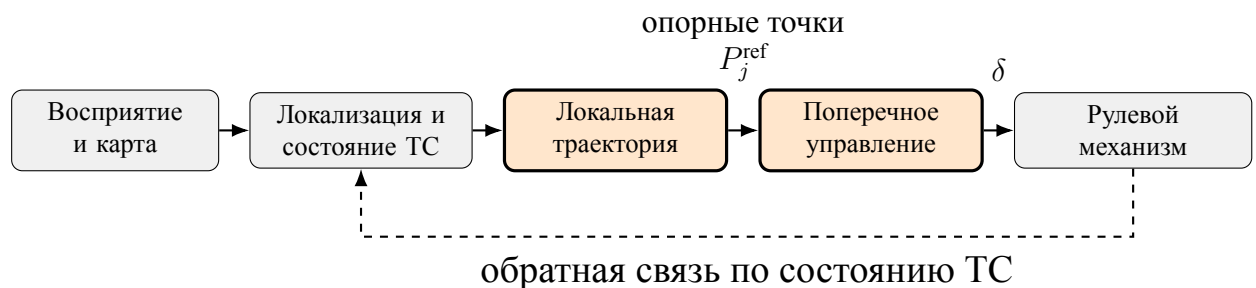


Рисунок 7 – Место задач главы 2 в функциональной архитектуре АСУД (выделены контуры, рассматриваемые в главе)

В рамках главы последовательно рассматриваются: представление модели в пространстве состояний, линеаризация, формирование моделей в координатах ошибок, синтез и сравнение алгоритмов следования траектории, а также воспроизводимый вычислительный эксперимент для выбора базового регулятора.

2.1 Модель автомобиля в пространстве состояний

В задачах управления движением беспилотного транспортного средства требуется описание динамики объекта управления в форме, допускающей многомерные входы и выходы, явный учет ограничений на состояния и управляющие воздействия, а также последующее применение методов оптимального и модельно-предиктивного управления. Указанные требования естественным образом обеспечиваются представлением динамики в *пространстве состояний*, в котором система задается уравнениями первого порядка относительно вектора внутренних переменных (переменных состояния) и алгебраическими соотношениями для выходных переменных.

Понятие состояния динамической системы относится к минимальному набору переменных $x_i(t)$, $i = 1, \dots, n$, знание которых в начальный момент времени t_0 совместно с заданием входных воздействий на интервале $t \geq t_0$ достаточно для прогнозирования будущего состояния и выходных переменных системы при $t > t_0$. Число n определяет порядок системы [57—59].

В стандартной форме математическое описание системы выражается набором из n связанных обыкновенных дифференциальных уравнений первого порядка (уравнений состояния), в которых производная по времени каждой переменной состояния выражается через переменные состояния и входные воздействия $u_j(t)$, $j = 1, \dots, r$. В общем случае система уравнений состояния имеет вид

$$\begin{aligned}\dot{x}_1(t) &= f_1(x, u, t), \\ \dot{x}_2(t) &= f_2(x, u, t), \\ &\vdots = \vdots \\ \dot{x}_n(t) &= f_n(x, u, t),\end{aligned}\tag{2.1}$$

где $\dot{x}_i = dx_i/dt$, а каждая функция $f_i(x, u, t)$, ($i = 1, \dots, n$), в общем случае является нелинейной и может зависеть от времени.

Уравнения (2.1) удобно представлять в векторной форме. Вектор состояния определяется как $x(t) = [x_1(t), x_2(t), \dots, x_n(t)]^T$, а вектор входов как $u(t) = [u_1(t), u_2(t), \dots, u_r(t)]^T$. Тогда система уравнений состояния записывается в виде

$$\dot{x} = f(x, u, t),\tag{2.2}$$

где $f(x, u, t)$ — вектор-функция с n компонентами.

Для линейной стационарной системы порядка n с r входами уравнения состояния (2.2) приводятся к линейной форме с постоянными коэффициентами

$$\dot{x}_i(t) = \sum_{j=1}^n a_{ij} x_j(t) + \sum_{k=1}^r b_{ik} u_k(t), \quad i = 1, \dots, n, \quad (2.3)$$

что в матричной записи эквивалентно

$$\dot{x}(t) = \begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \vdots \\ \dot{x}_n(t) \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_n(t) \end{bmatrix} + \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1r} \\ b_{21} & b_{22} & \dots & b_{2r} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n2} & \dots & b_{nr} \end{bmatrix} \begin{bmatrix} u_1(t) \\ u_2(t) \\ \vdots \\ u_r(t) \end{bmatrix}. \quad (2.4)$$

Обобщенная запись (2.4) имеет вид

$$\dot{x} = Ax + Bu, \quad (2.5)$$

где $x \in \mathbb{R}^n$, $u \in \mathbb{R}^r$, $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times r}$.

Выходными переменными системы считаются величины, представляющие интерес в контексте рассматриваемой задачи управления. Описание физической системы в терминах набора переменных состояния не обязательно включает все величины, представляющие непосредственный инженерный интерес. Для линейной системы произвольная выходная переменная может быть представлена линейной комбинацией переменных состояния и входных воздействий

$$y(t) = \sum_{j=1}^n c_j x_j(t) + \sum_{k=1}^r d_k u_k(t), \quad (2.6)$$

где c_j и d_k — постоянные коэффициенты. Если задано m выходных переменных, то уравнения выходов записываются в виде

$$\begin{bmatrix} y_1(t) \\ y_2(t) \\ \vdots \\ y_m(t) \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & \dots & c_{1n} \\ c_{21} & c_{22} & \dots & c_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{m1} & c_{m2} & \dots & c_{mn} \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_n(t) \end{bmatrix} + \begin{bmatrix} d_{11} & d_{12} & \dots & d_{1r} \\ d_{21} & d_{22} & \dots & d_{2r} \\ \vdots & \vdots & \ddots & \vdots \\ d_{m1} & d_{m2} & \dots & d_{mr} \end{bmatrix} \begin{bmatrix} u_1(t) \\ u_2(t) \\ \vdots \\ u_r(t) \end{bmatrix}. \quad (2.7)$$

В компактной форме (2.7) записывается как

$$y = Cx + Du, \quad (2.8)$$

где $y \in \mathbb{R}^m$, $C \in \mathbb{R}^{m \times n}$, $D \in \mathbb{R}^{m \times r}$. Для многих физических систем матрица D является нулевой, и тогда (2.8) упрощается до

$$y = Cx. \quad (2.9)$$

Полная модель линейной стационарной системы задается уравнениями состояния (2.5) и уравнениями выходов (2.8)

$$\dot{x} = Ax + Bu, \quad (2.10)$$

$$y = Cx + Du. \quad (2.11)$$

Матрицы A и B определяются структурой объекта управления и его параметрами, тогда как матрицы C и D определяются выбором выходных переменных в соответствии с постановкой задачи. В дальнейших разделах данное представление используется при выводе и линеаризации моделей движения автомобиля и при синтезе алгоритмов следования траектории.

2.2 Линеаризация с помощью ряда Тейлора

Линеаризация применяется для получения локально-линейного приближения нелинейной модели в окрестности выбранной рабочей точки или опорной траектории. Такой переход используется для упрощения анализа и для применения методов линейной теории управления, а также при построении локальных моделей в задачах модельно-предиктивного управления [60—62].

Рассмотрим нелинейную динамическую систему, заданную в пространстве состояний:

$$\dot{x} = f(x, u, t), \quad (2.12)$$

где $x(t) \in \mathbb{R}^n$ — вектор состояния, $u(t) \in \mathbb{R}^r$ — вектор управления, $f(\cdot)$ — вектор-функция, в общем случае нелинейная и зависящая от времени. При

необходимости дополнительно вводится выходное уравнение:

$$y = g(x, u, t), \quad (2.13)$$

где $y(t) \in \mathbb{R}^m$ — вектор выходных величин.

Пусть задана рабочая точка (точка разложения) (x^*, u^*) , соответствующая фиксированному режиму движения, либо опорная траектория $(x_{\text{ref}}(t), u_{\text{ref}}(t))$, задающая требуемое движение. Введем отклонения

$$\delta x = x - x^*, \quad \delta u = u - u^*, \quad (2.14)$$

или, в случае опорной траектории,

$$\delta x = x - x_{\text{ref}}(t), \quad \delta u = u - u_{\text{ref}}(t). \quad (2.15)$$

Разложение правой части (2.12) в ряд Тейлора по переменным (x, u) с удержанием членов первого порядка приводит к локальному приближению:

$$f(x, u, t) \approx f(x^*, u^*, t) + A(t) \delta x + B(t) \delta u, \quad (2.16)$$

где матрицы $A(t)$ и $B(t)$ определяются якобианами:

$$A(t) = \left. \frac{\partial f}{\partial x} \right|_{x=x^*, u=u^*}, \quad B(t) = \left. \frac{\partial f}{\partial u} \right|_{x=x^*, u=u^*}. \quad (2.17)$$

Аналогично для выходного уравнения (2.13):

$$g(x, u, t) \approx g(x^*, u^*, t) + C(t) \delta x + D(t) \delta u, \quad (2.18)$$

где

$$C(t) = \left. \frac{\partial g}{\partial x} \right|_{x=x^*, u=u^*}, \quad D(t) = \left. \frac{\partial g}{\partial u} \right|_{x=x^*, u=u^*}. \quad (2.19)$$

В явном виде матрица Якоби по состоянию для вектор-функции

$f = [f_1, \dots, f_n]^T$ имеет вид

$$\frac{\partial f}{\partial x} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}. \quad (2.20)$$

Подстановка (2.16) в (2.12) дает аффинную локально-линейную модель. В терминах отклонений она может быть записана как

$$\delta \dot{x} \approx A(t) \delta x + B(t) \delta u + w(t), \quad (2.21)$$

где

$$w(t) = f(x^*, u^*, t) - \dot{x}^*. \quad (2.22)$$

Если (x^*, u^*) соответствует равновесию, то $\dot{x}^* = 0$, а при выборе опорной траектории, согласованной с исходной моделью, выполняется $\dot{x}_{\text{ref}}(t) = f(x_{\text{ref}}(t), u_{\text{ref}}(t), t)$ и добавка $w(t)$ обращается в нуль. В этом случае получаем стандартную линейную модель в отклонениях:

$$\delta \dot{x} \approx A(t) \delta x + B(t) \delta u, \quad (2.23)$$

а для выходов

$$\delta y \approx C(t) \delta x + D(t) \delta u. \quad (2.24)$$

Линейное приближение (2.23)–(2.24) корректно описывает динамику в окрестности рабочей точки; погрешность аппроксимации является величиной второго и более высоких порядков по норме отклонений $\|\delta x\|$ и $\|\delta u\|$. При существенном удалении от точки разложения точность модели ухудшается, что требует либо применения нелинейных методов управления, либо рекурсивного обновления рабочей точки и матриц Якоби. Подход с обновлением линейной модели на каждом шаге характерен для алгоритмов модельно-предиктивного управления и для расширенного фильтра Калмана [61, 63].

Геометрическая интерпретация линеаризации одномерной функции в окрестности точки разложения иллюстрируется на рисунке 8. Синий график соответствует исходной функции, оранжевый график соответствует ее линейной аппроксимации в окрестности точки x^*

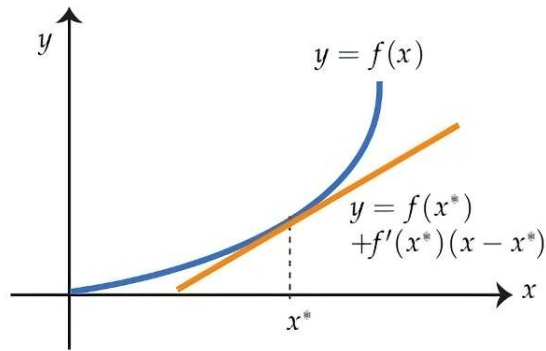


Рисунок 8 – Линеаризация с помощью ряда Тейлора в окрестности точки разложения. Синий график — исходная функция. Оранжевый график — линейная аппроксимация. x^* — точка разложения.

Полученная локально-линейная модель используется в настоящей работе при выводе и анализе моделей движения автомобиля, а также при построении регуляторов и оптимизационных постановок, в которых требуется явная матричная форма уравнений состояния и возможность учета ограничений [45, 60].

2.3 Математические модели автомобиля

2.3.1 Кинематическая велосипедная модель

Состояние и управление зададим в виде

$$\mathbf{x} = \begin{bmatrix} x \\ y \\ \psi \\ v \end{bmatrix}, \quad \mathbf{u} = \begin{bmatrix} a \\ \delta \end{bmatrix}, \quad (2.25)$$

где (x, y) — координаты центра задней оси автомобиля в глобальной системе координат, ψ — угол рысканья, v — продольная скорость, a — продольное ускорение, δ — угол поворота управляемых колес. Колесная база обозначается $L = L_f + L_r$.

Расчетная схема кинематической модели представлена на рисунке 9.

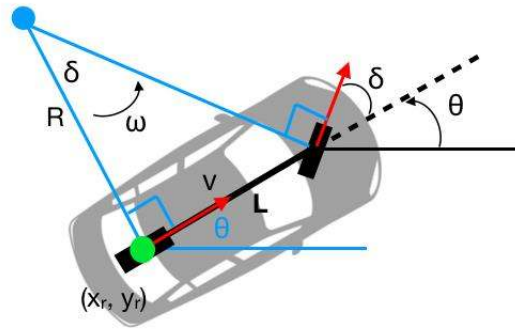


Рисунок 9 – Расчётная схема кинематической велосипедной модели с опорной точкой в центре задней оси автомобиля

Уравнения движения в непрерывной форме имеют вид

$$\begin{aligned}\dot{x} &= v \cos \theta, \\ \dot{y} &= v \sin \theta, \\ \dot{\theta} &= \frac{v}{L} \tan \delta, \\ \dot{v} &= a.\end{aligned}\tag{2.26}$$

Векторная форма (2.26):

$$f(\mathbf{x}, \mathbf{u}) = \begin{bmatrix} v \cos \theta \\ v \sin \theta \\ \frac{v}{L} \tan \delta \\ a \end{bmatrix}.\tag{2.27}$$

Линеаризуем (2.27) в окрестности опорных значений $(\mathbf{x}_{\text{ref}}, \mathbf{u}_{\text{ref}})$, вводя отклонения

$$\delta \mathbf{x} = \mathbf{x} - \mathbf{x}_{\text{ref}}, \quad \delta \mathbf{u} = \mathbf{u} - \mathbf{u}_{\text{ref}}.\tag{2.28}$$

Для продольной компоненты опорного состояния здесь и далее используется единое обозначение $\bar{v} = v^{\text{ref}}$. Матрицы Якоби равны

$$A = \left. \frac{\partial f}{\partial \mathbf{x}} \right|_{(\mathbf{x}_{\text{ref}}, \mathbf{u}_{\text{ref}})}, \quad B = \left. \frac{\partial f}{\partial \mathbf{u}} \right|_{(\mathbf{x}_{\text{ref}}, \mathbf{u}_{\text{ref}})},\tag{2.29}$$

и в явном виде записываются как

$$A = \begin{bmatrix} 0 & 0 & -\bar{v} \sin \theta_{\text{ref}} & \cos \theta_{\text{ref}} \\ 0 & 0 & \bar{v} \cos \theta_{\text{ref}} & \sin \theta_{\text{ref}} \\ 0 & 0 & 0 & \frac{1}{L} \tan \delta_{\text{ref}} \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & \frac{\bar{v}}{L \cos^2 \delta_{\text{ref}}} \\ 1 & 0 \end{bmatrix}. \quad (2.30)$$

Тогда локально-линейная модель в отклонениях принимает вид

$$\delta \dot{\mathbf{x}} = A \delta \mathbf{x} + B \delta \mathbf{u}. \quad (2.31)$$

Для малых углов $\theta_{\text{ref}} \approx 0$, $\delta_{\text{ref}} \approx 0$ выражения (2.30) упрощаются до

$$A \approx \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & \bar{v} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad B \approx \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & \frac{\bar{v}}{L} \\ 1 & 0 \end{bmatrix}. \quad (2.32)$$

2.3.2 Динамическая велосипедная модель

Основным отличием динамической велосипедной модели от кинематической является учет бокового увода колес. Расчетная схема представлена на рисунке 10.

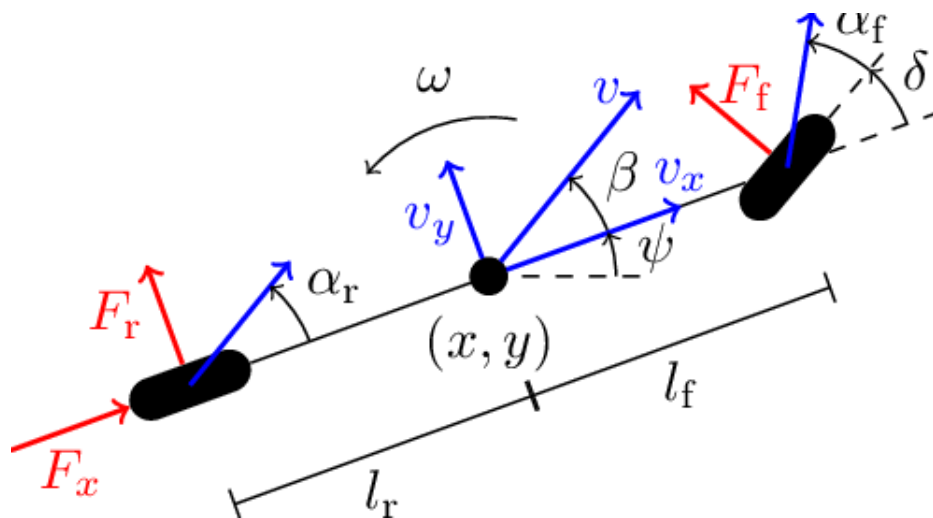


Рисунок 10 – Расчетная схема динамической велосипедной модели с опорной точкой в центре масс автомобиля

В обозначениях рисунка 10 координаты (x, y) соответствуют центру масс автомобиля, а $\omega \equiv r = \dot{\psi}$ обозначает угловую скорость рыскания. Продольная сила F_x , показанная на расчетной схеме, в модели (2.35) обобщенно учитывается посредством входного продольного ускорения a_x ; отдельная модель продольного привода в настоящем подразделе не рассматривается.

Состояние и входы описываются как

$$\mathbf{x} = \begin{bmatrix} v_x \\ v_y \\ \psi \\ r \end{bmatrix}, \quad \mathbf{u} = \begin{bmatrix} a_x \\ \delta \end{bmatrix}. \quad (2.33)$$

Линейная модель сил шин:

$$\begin{aligned} F_f &= C_f \left(\delta - \beta - \frac{l_f r}{v_x} \right), \\ F_r &= C_r \left(-\beta + \frac{l_r r}{v_x} \right), \\ \beta &= \arctan \left(\frac{v_y}{v_x} \right). \end{aligned} \quad (2.34)$$

Уравнения движения (плоская динамика Ньютона–Эйлера):

$$\begin{aligned} \dot{v}_x &= a_x + v_y r, \\ \dot{v}_y &= \frac{1}{m} (F_f \cos \delta + F_r) - v_x r, \\ \dot{r} &= \frac{1}{I_z} (l_f F_f \cos \delta - l_r F_r), \\ \dot{\psi} &= r. \end{aligned} \quad (2.35)$$

Здесь m — масса автомобиля, I_z — момент инерции относительно вертикальной оси, l_f и l_r — расстояния от центра масс до передней и задней осей, C_f и C_r — суммарные боковые жесткости соответствующих осей. Модель (2.35) описывает полный объект, поэтому содержит продольную скорость v_x как состояние и продольное ускорение a_x как вход. В рассматриваемом далее поперечном QR-регуляторе эти величины не являются переменными решения: заданный скоростной профиль отслеживается отдельным продольным контуром.

2.3.3 Линеаризованная модель поперечной динамики и ошибок слежения

Рассматриваемый регулятор решает задачу поперечного управления. Продольный профиль движения формируется внешним модулем планирования до вызова регулятора и передается ему в составе последовательности опорных точек

$$P_j^{\text{ref}} = (X_j^{\text{ref}}, Y_j^{\text{ref}}, \psi_j^{\text{ref}}, \kappa_j^{\text{ref}}, v_j^{\text{ref}}). \quad (2.36)$$

На шаге управления k известная опорная скорость обозначается как

$$\bar{v}_{k+i} = v_{k+i}^{\text{ref}}, \quad r_{k+i}^{\text{ref}} = \bar{v}_{k+i} \kappa_{k+i}^{\text{ref}}, \quad i = 0, \dots, N_p, \quad (2.37)$$

где r^{ref} — требуемая угловая скорость рыскания. Размерности величин $\bar{v}$, κ^{ref} и r^{ref} составляют соответственно м/с, м⁻¹ и с⁻¹. Первые N_p точек параметризуют интервалы управления, а точка с индексом $k + N_p$ задает параметры конечного состояния. Величины $\bar{v}_{k+i}$ и $\kappa_{k+i}^{\text{ref}}$ не входят в вектор состояния и вектор оптимизируемых переменных поперечного регулятора и не изменяются QR-решателем. Слежение за заданным продольным профилем обеспечивается отдельным контуром управления.

При выводе модели приняты следующие допущения: автомобиль представлен одноколейной динамической схемой; боковые реакции шин линейно зависят от углов увода; угол поворота колес, углы увода, курсовая и поперечная ошибки малы; фактическая продольная скорость в поперечной модели приближенно равна заданной, то есть $v_x \approx \bar{v}$; величины $\bar{v}$ и r^{ref} постоянны внутри одного интервала дискретизации; параметры m , I_z , l_f , l_r , C_f и C_r постоянны. Здесь и далее C_f и C_r обозначают суммарные боковые жесткости передней и задней осей с размерностью Н·рад⁻¹. Если исходные параметры c_f и c_r заданы для одной шины, их преобразование выполняется один раз:

$$C_f = 2c_f, \quad C_r = 2c_r. \quad (2.38)$$

В качестве исходной принимается модель в физических координатах

$$\boldsymbol{\xi} = \begin{bmatrix} e_y & e_\psi & v_y & r \end{bmatrix}^T, \quad (2.39)$$

где e_y и e_ψ — поперечная и курсовая ошибки, v_y — боковая скорость центра масс,

$r = \dot{\psi}$ — угловая скорость рыскания. Единственным управляющим воздействием поперечного регулятора является угол поворота передних управляемых колес δ . Для малой курсовой ошибки кинематические связи имеют вид

$$\dot{e}_y = v_y + \bar{v}e_\psi, \quad \dot{e}_\psi = r - r^{\text{ref}}, \quad r^{\text{ref}} = \bar{v}\kappa^{\text{ref}}. \quad (2.40)$$

С учетом линейной модели шин уравнения поперечной и рыскательной динамики записываются следующим образом:

$$\begin{aligned} m(\dot{v}_y + \bar{v}r) &= C_f\delta - C_f\frac{v_y + l_fr}{\bar{v}} - C_r\frac{v_y - l_rr}{\bar{v}}, \\ I_z\dot{r} &= l_fC_f\delta - l_fC_f\frac{v_y + l_fr}{\bar{v}} + l_rC_r\frac{v_y - l_rr}{\bar{v}}. \end{aligned} \quad (2.41)$$

После объединения (2.40) и (2.41) физическая модель принимает вид

$$\dot{\boldsymbol{\xi}} = A_p(\bar{v})\boldsymbol{\xi} + B_p\delta + E_pr^{\text{ref}}, \quad (2.42)$$

где

$$A_p(\bar{v}) = \begin{bmatrix} 0 & \bar{v} & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\frac{C_f + C_r}{m\bar{v}} & \frac{C_rl_r - C_fl_f}{m\bar{v}} - \bar{v} \\ 0 & 0 & \frac{C_rl_r - C_fl_f}{I_z\bar{v}} & -\frac{C_fl_f^2 + C_rl_r^2}{I_z\bar{v}} \end{bmatrix}, \quad (2.43)$$

$$B_p = \begin{bmatrix} 0 \\ 0 \\ \frac{C_f}{m} \\ \frac{C_fl_f}{I_z} \end{bmatrix}, \quad E_p = \begin{bmatrix} 0 \\ -1 \\ 0 \\ 0 \end{bmatrix}. \quad (2.44)$$

Таким образом, $\bar{v} = v^{\text{ref}}$ является известным кусочно-постоянным параметром модели, а не переменной оптимизации. До подстановки конкретного скоростного профиля (2.42) представляет собой линейную модель с параметрами, зависящими от скорости.

Для синтеза LQR- и MPC-регуляторов далее используется эквивалентный

вектор состояния в координатах производных ошибок

$$\mathbf{x}_e = \begin{bmatrix} e_y & \dot{e}_y & e_\psi & \dot{e}_\psi \end{bmatrix}^T. \quad (2.45)$$

Из (2.40) следуют обратные соотношения

$$v_y = \dot{e}_y - \bar{v}e_\psi, \quad r = \dot{e}_\psi + r^{\text{ref}}. \quad (2.46)$$

При фиксированных на интервале значениях $\bar{v}$ и r^{ref} преобразование координат имеет вид

$$\mathbf{x}_e = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \bar{v} & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \boldsymbol{\xi} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ -r^{\text{ref}} \end{bmatrix}. \quad (2.47)$$

При допущениях $\dot{\bar{v}} = 0$ и $\dot{r}^{\text{ref}} = 0$ внутри интервала дискретизации преобразованная модель записывается как

$$\dot{\mathbf{x}}_e = A_e(\bar{v})\mathbf{x}_e + B_e\delta + G_e(\bar{v})r^{\text{ref}}, \quad (2.48)$$

где

$$A_e(\bar{v}) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{C_f + C_r}{m\bar{v}} & \frac{C_f + C_r}{m} & \frac{C_rl_r - C_fl_f}{m\bar{v}} \\ 0 & 0 & 0 & 1 \\ 0 & \frac{C_rl_r - C_fl_f}{I_z\bar{v}} & \frac{C_fl_f - C_rl_r}{I_z} & -\frac{C_fl_f^2 + C_rl_r^2}{I_z\bar{v}} \end{bmatrix}, \quad (2.49)$$

$$B_e = \begin{bmatrix} 0 \\ \frac{C_f}{m} \\ 0 \\ \frac{C_fl_f}{I_z} \end{bmatrix}, \quad G_e(\bar{v}) = \begin{bmatrix} 0 \\ \frac{C_rl_r - C_fl_f}{m\bar{v}} - \bar{v} \\ 0 \\ -\frac{C_fl_f^2 + C_rl_r^2}{I_z\bar{v}} \end{bmatrix}. \quad (2.50)$$

Различие матриц $A_p(\bar{v})$ и $A_e(\bar{v})$ обусловлено заменой физических переменных v_y и r производными ошибок $\dot{e}_y$ и $\dot{e}_\psi$, а не использованием разных моделей автомобиля. В частности, при вычислении $\ddot{e}_y = \dot{v}_y + \bar{v}\dot{e}_\psi$ слагаемые $-\bar{v}r$ и $+\bar{v}r$ взаимно сокращаются. Поэтому элемент $A_{e,24}$ не содержит слагаемого $-\bar{v}$;

указанное слагаемое сохраняется во втором элементе $G_e(\bar{v})$.

Наличие элементов $1/\bar{v}$ ограничивает область применения динамической модели условием

$$\bar{v} \geq v_{\min}, \quad v_{\min} = 3 \text{ м/с}. \quad (2.51)$$

В принятой архитектуре ограничение (2.51) обеспечивается внешним модулем формирования продольного профиля: опорные точки с $\bar{v} < v_{\min}$ не передаются рассматриваемому динамическому LQR/MPC-регулятору. Режим трогания и полной остановки относится к отдельному низкоскоростному контуру и в настоящую постановку поперечного QR не входит. Таким образом, во всех рассматриваемых вычислительных экспериментах $\bar{v} \geq v_{\min} = 3 \text{ м/с}$. После подстановки последовательности (2.37) модель на горизонте прогнозирования является линейной нестационарной моделью с известными матрицами.

2.3.4 Кинематическая велосипедная модель в фрейме Френе

Для описания поперечной динамики относительно опорной траектории $\mathcal{C}$ используем кинематическую модель велосипеда, записанную в криволинейных координатах Френе [29, 64].

Далее считаем, что $(\cdot)' = \frac{d(\cdot)}{ds}$ — производная по продольной координате s (по дуговой длине референсной кривой).

Связь между производной по времени и производной по дуговой координате задается соотношением $\dot{z} = z' \dot{s}$. Для кинематической модели справедливо

$$\dot{s} = \frac{v \cos(e_\theta)}{1 - \kappa^{\text{ref}}(s) d}, \quad (2.52)$$

что позволяет при необходимости переходить от модели по s к модели по времени при дискретизации и реализации в контуре управления.

Введем вектор состояния

$$z = \begin{bmatrix} d & e_\theta & \kappa \end{bmatrix}^T, \quad (2.53)$$

где d — поперечное смещение (латеральная ошибка), e_θ — угловая ошибка (разность между фактическим углом курса и направлением касательной к опорной траектории), κ — кривизна траектории движения транспортного средства, а $\kappa^{\text{ref}}(s)$ — кривизна опорной траектории.

В качестве управления рассматривается скорость изменения кривизны по s :
 $u = \kappa'$.

Тогда поперечная кинематическая модель во фрейме Френе имеет вид

$$z' = \begin{bmatrix} d' \\ e'_\theta \\ \kappa' \end{bmatrix} = \begin{bmatrix} (1 - \kappa^{\text{ref}} d) \tan(e_\theta) \\ \frac{(1 - \kappa^{\text{ref}} d) \kappa}{\cos(e_\theta)} - \kappa^{\text{ref}} \\ u \end{bmatrix}. \quad (2.54)$$

Линеаризуем (2.54) в окрестности референсного состояния

$$d = 0, \quad e_\theta = 0, \quad \kappa = \kappa^{\text{ref}}(s), \quad (2.55)$$

и опорного управления $u^{\text{ref}} = (\kappa^{\text{ref}})'(s)$. Введем отклонения

$$\tilde{z} = \begin{bmatrix} d & e_\theta & \tilde{\kappa} \end{bmatrix}^T, \quad \tilde{\kappa} = \kappa - \kappa^{\text{ref}}, \quad \tilde{u} = u - (\kappa^{\text{ref}})'. \quad (2.56)$$

При малых ошибках ($\tan e_\theta \approx e_\theta$, $\cos e_\theta \approx 1$) получаем линейную модель в отклонениях

$$\tilde{z}' = \underbrace{\begin{bmatrix} 0 & 1 & 0 \\ -(\kappa^{\text{ref}})^2 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}}_{A(s)} \tilde{z} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}}_B \tilde{u}, \quad (2.57)$$

где $\kappa^{\text{ref}} = \kappa^{\text{ref}}(s)$ вычисляется в соответствующей опорной точке.

Модель (2.57) используется далее как предсказательная (prediction model) при постановке задачи оптимизации траектории.

Для обеспечения реализуемости траектории вводятся ограничения на κ и $u = \kappa'$.

На низких скоростях ограничивающим фактором является кинематика рулевого управления: максимальный угол поворота управляемых колес δ_{max} [64, 65] задает предел кривизны

$$|\kappa| \leq \kappa_{\text{geom}} = \frac{\tan(\delta_{\text{max}})}{L}, \quad (2.58)$$

где L — колесная база.

На высоких скоростях доминируют динамические ограничения по боковому

ускорению [65—67]:

$$|a_y| = v^2 |\kappa| \leq a_{y,\max}, \quad a_{y,\max} \approx \mu g, \quad (2.59)$$

откуда

$$|\kappa| \leq \kappa_{\text{dyn}} = \frac{\mu g}{v^2}. \quad (2.60)$$

Типичные диапазоны коэффициента сцепления шин с покрытием для асфальта составляют порядка $\mu \approx 0.8\text{--}0.9$ (сухое покрытие) и $\mu \approx 0.5\text{--}0.7$ (влажное покрытие), поэтому в задачах планирования целесообразно выбирать консервативные значения μ в зависимости от условий движения.

Итоговое ограничение на кривизну удобно задавать в виде двухрежимной (скоростезависимой) границы:

$$|\kappa| \leq \kappa_{\max}(v) = \min\left(\frac{\tan(\delta_{\max})}{L}, \frac{\mu g}{v^2}\right). \quad (2.61)$$

Дополнительно ограничим скорость изменения кривизны

$$|u| = |\kappa'| \leq u_{\max}, \quad (2.62)$$

что обеспечивает ограничение на резкость маневра и улучшает отслеживаемость траектории регулятором (в частности, снижает требования к скорости рулевого привода).

Таким образом, кинематическая модель в координатах Френе и введенные ограничения позволяют формализовать задачу генерации траектории как задачу оптимизации с линейной динамикой (после линеаризации) и простыми ограничениями на состояние и управление.

Дальнейшее рассмотрение ограничено режимами, соответствующими штатному городскому движению: малые углы увода, отсутствие срыва сцепления, отсутствие комбинированного скольжения и экстремальных маневров. Моделирование заноса/дрифта и предельной нелинейности шин требует использования нелинейных моделей шин и расширенных моделей динамики, что выходит за рамки целей настоящей работы.

2.4 Аппроксимация опорной траектории кубическим В-сплайном

Для формирования опорной траектории, пригодной для исполнения в контуре управления, требуется обеспечить непрерывное изменение ее геометрических характеристик. В настоящей работе опорный путь аппроксимируется равномерным кубическим В-сплайном. Такое представление обеспечивает непрерывность координатной функции и ее первых двух производных, а также непрерывность кривизны на регулярных участках кривой. Кривизна и необходимые для дальнейших расчетов производные вычисляются аналитически внутри каждого сплайнового сегмента.

Пусть P_i — контрольные точки, а параметр $t \in [0, 1]$ задает положение внутри сплайнового сегмента. Тогда точка на кубическом В-сплайне выражается как

$$P(t) = \begin{bmatrix} 1 & t & t^2 & t^3 \end{bmatrix} \frac{1}{6} \begin{bmatrix} 1 & 4 & 1 & 0 \\ -3 & 0 & 3 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1 \end{bmatrix} \begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{bmatrix}. \quad (2.63)$$

Кривизна плоской параметрической кривой $P(t) = (P_x(t), P_y(t))$ вычисляется по формуле

$$\kappa(t) = \frac{\begin{vmatrix} P'_x(t) & P''_x(t) \\ P'_y(t) & P''_y(t) \end{vmatrix}}{\|P'(t)\|^3}, \quad (2.64)$$

где штрих обозначает производную по параметру t . Вычисленные значения опорной кривизны во всех последующих моделях обозначаются единым символом κ^{ref} . Пример кубического В-сплайна и распределение кривизны представлено на рисунке 11.

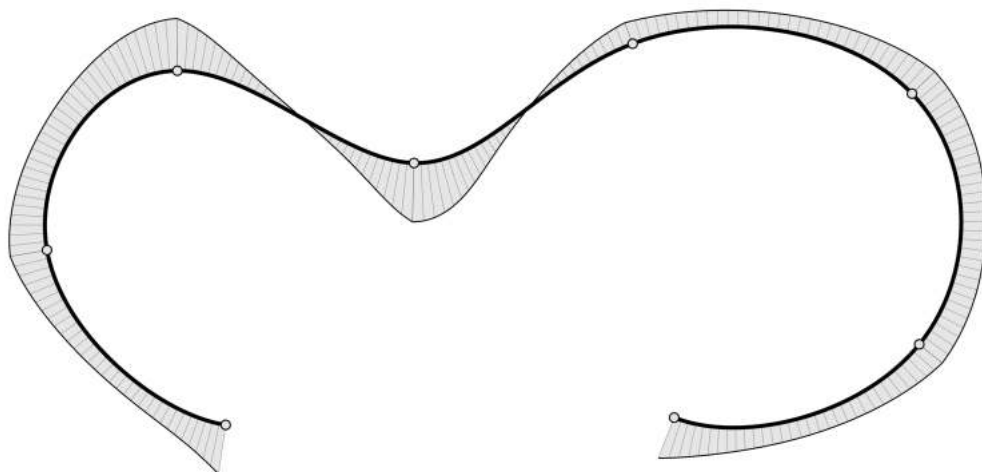


Рисунок 11 – Пример кубического В-сплайна и распределение кривизны

В дальнейших главах это представление будет использовано для оптимизации параметров траектории с учетом динамики транспортного средства и построения адаптивного регулятора, способного работать в реальном времени.

2.5 Алгоритмы следования траектории

Среди различных методов управления, модельное прогнозирующее управление (MPC), линейно-квадратичный регулятор (LQR) и Pure Pursuit стали популярными подходами из-за их отличных характеристик и областей применения.

MPC широко известен своей способностью оптимизировать управляющие действия с учетом ограничений системы и будущих состояний. Это делает его идеальным для управления сложной динамикой транспортного средства, например, при планировании маршрута и объезде препятствий. Его реализация основана на динамической линейной модели процесса. В контрольной временной области MPC в основном оптимизирует текущий момент, но также учитывает будущие моменты для получения оптимального решения в текущий момент, а затем многократно оптимизирует для достижения оптимальных решений во всей временной области.

С другой стороны, LQR - это хорошо зарекомендовавший себя метод управления, известный своей простотой и вычислительной эффективностью, обеспечивающий оптимальные решения для линейных систем без ограничений. LQR (линейный квадратичный регулятор) - это метод проектирования

пространства состояний, который является очень развитым в современной теории управления. Алгоритм LQR позволяет получить закон оптимального управления с линейной обратной связью по состоянию и легко формирует оптимальное управление с замкнутым контуром. Оптимальная конструкция LQR означает, что разработанное устройство обратной связи по состоянию K должно минимизировать квадратичную целевую функцию J , а K однозначно определяется взвешенными матрицами Q и R . При разных скоростях автомобиля различия между Q и R будут оказывать существенное влияние на производительность алгоритма, поэтому выбор Q и R очень важен.

Pure Pursuit - это алгоритм геометрического отслеживания траектории, который часто используется в автономной навигации транспортных средств благодаря его простоте реализации и эффективности в приложениях реального времени.

Каждый из этих методов управления обладает уникальными преимуществами и ограничениями, что делает их подходящими для различных аспектов управления транспортным средством. MPC отлично подходит для сценариев, где необходимы ограничения и возможности прогнозирования, в то время как LQR обеспечивает более быстрое время отклика в более простых системах. Чистое преследование, хотя и эффективно для отслеживания траектории, может вызвать трудности в более сложных сценариях, особенно на высоких скоростях или при резких изменениях траектории.

В современной теории управления модельное прогнозирующее управление (MPC) обычно фокусируется на линейных системах, представленных уравнениями в пространстве состояний. Однако MPC также может применяться к нелинейным системам, что делает его универсальным подходом к управлению. Напротив, линейный квадратичный регулятор (LQR) предназначен для линейных систем, также представленных в виде пространства состояний. Для повышения эффективности и сложности вычислений нелинейные системы часто линеаризуются для анализа в рамках LQR. Как MPC, так и LQR используют модель динамики велосипеда в качестве исследуемой системы. Хотя модель велосипеда по своей сути нелинейна, оба контроллера нацелены на поиск оптимальных решений для управления путем линеаризации системы в процессе оптимизации. В результате, несмотря на нелинейный характер динамики транспортного средства, системы, эффективно

рассматриваются как линейные для целей оптимизации управления.

2.5.1 Принцип работы алгоритма Чистое преследование (Pure Pursuit, PP)

Алгоритм Pure Pursuit относится к геометрическим методам поперечного управления и не требует явного использования динамической модели. В качестве опорной точки, характеризующей текущее положение транспортного средства, принимается центр его задней оси. На каждом такте управления выбирается целевая точка на опорной траектории на расстоянии взгляда вперед l_d , отсчитываемом от центра задней оси автомобиля. Затем определяется управляющее воздействие на рулевое управление, обеспечивающее движение транспортного средства по дуге окружности, проходящей через центр задней оси и выбранную целевую точку (рисунок 12).

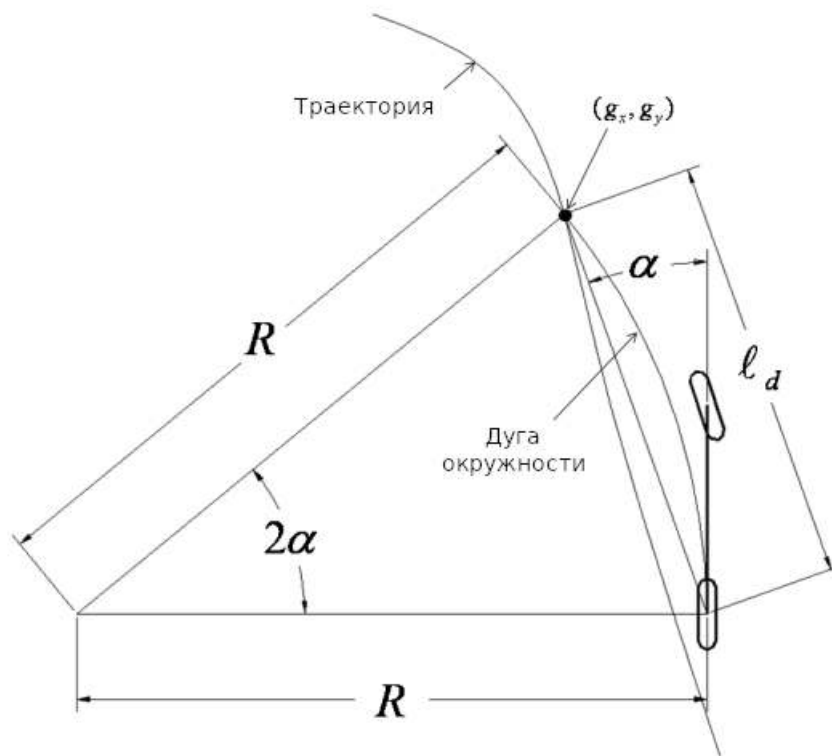


Рисунок 12 – Геометрическая интерпретация алгоритма Pure Pursuit

Команда по углу поворота управляемых колес задается выражением

$$\delta(t) = \tan^{-1} \left(\frac{2L \sin(\alpha(t))}{l_d(v(t))} \right), \quad (2.65)$$

где L — колесная база, $\alpha(t)$ — угловая ошибка на целевую точку относительно текущего курса.

Ключевым параметром является расстояние взгляда вперед l_d , которое в практических реализациях целесообразно выбирать как функцию скорости для обеспечения компромисса между точностью и устойчивостью:

$$l_d(v) = \text{sat}_{[l_{\min}, l_{\max}]}(l_0 + k_v v), \quad (2.66)$$

где $l_0, k_v, l_{\min}, l_{\max}$ — настраиваемые параметры, $\text{sat}_{[a,b]}(\cdot)$ — операция насыщения по интервалу.

Для обеспечения физической реализуемости управляющего воздействия вводятся ограничения по углу руления и по скорости изменения угла руления. Ограничение по углу задается как

$$-\delta_{\max} \leq \delta(t) \leq \delta_{\max}, \quad (2.67)$$

а ограничение по скорости изменения угла руления в дискретном времени с шагом T_s удобно задавать в виде

$$-\dot{\delta}_{\max} T_s \leq \delta_k - \delta_{k-1} \leq \dot{\delta}_{\max} T_s. \quad (2.68)$$

Следует отметить, что при увеличении скорости и/или при движении по кривым малого радиуса качество слежения РР может деградировать. Это обусловлено тем, что требуемая для прохождения криволинейного участка величина поперечного ускорения пропорциональна квадрату скорости:

$$a_y = v^2 \kappa, \quad (2.69)$$

где κ — кривизна траектории. В РР кривизна формируемой дуги в первом приближении определяется геометрией на целевую точку и величиной l_d , поэтому при росте v требуемое a_y возрастает как v^2 , что приводит к частым режимам насыщения по δ и/или $\dot{\delta}$. В результате увеличивается поперечная ошибка e_y на кривых малого радиуса и возрастает чувствительность к выбору $l_d(v)$.

2.5.2 Принцип работы линейно-квадратичного регулятора следования траектории

Задача поперечного управления при следовании опорной траектории сводится к формированию такого угла поворота управляемых колес, при котором минимизируются поперечное отклонение автомобиля от опорной траектории, ошибка ориентации относительно касательной к траектории, а также скорости изменения указанных ошибок. Для построения регулятора используется единая линеаризованная модель (2.48), полученная из физической модели (2.42) посредством преобразования координат (2.47).

Вектор состояния принят в виде

$$\mathbf{x}_e = \begin{bmatrix} e_y & \dot{e}_y & e_\psi & \dot{e}_\psi \end{bmatrix}^T, \quad (2.70)$$

где e_y — поперечная ошибка, $\dot{e}_y$ — скорость изменения поперечной ошибки, e_ψ — ошибка по курсу, $\dot{e}_\psi$ — скорость изменения ошибки по курсу.

В непрерывном времени модель ошибок записывается в виде

$$\dot{\mathbf{x}}_e = A_e(\bar{v})\mathbf{x}_e + B_e\delta + G_e(\bar{v})r^{\text{ref}}, \quad (2.71)$$

где δ — угол поворота управляемых колес, а известное задающее воздействие определяется как $r^{\text{ref}} = \bar{v}\kappa^{\text{ref}}$. Продольная скорость не является управляющим воздействием LQR и не изменяется данным регулятором.

Таким образом, при синтезе управления кривизна и опорная скорость рассматриваются как известные параметры, а задача регулирования сводится к подавлению отклонений состояния $\mathbf{x}_e$ от нулевого значения.

Для реализации регулятора в цифровом вычислителе используется локальная модель с опорными параметрами, замороженными на текущем интервале. При дискретизации явным методом Эйлера с шагом T_s она имеет вид

$$\mathbf{x}_{e,k+1}^- = A_d^e(\bar{v}_k)\mathbf{x}_{e,k} + B_d^e\delta_k + \mathbf{g}_{d,k}^e, \quad (2.72)$$

где

$$A_d^e(\bar{v}_k) = I + T_s A_e(\bar{v}_k), \quad B_d^e = T_s B_e, \quad \mathbf{g}_{d,k}^e = T_s G_e(\bar{v}_k) r_k^{\text{ref}}. \quad (2.73)$$

Знак «минус» в $\mathbf{x}_{e,k+1}^-$ обозначает состояние до перехода к параметрам следующей опорной точки. На каждом новом такте $\mathbf{x}_{e,k+1}$ повторно формируется по измерениям и обновленным значениям $\bar{v}_{k+1}$ и r_{k+1}^{ref} . Известное смещение $\mathbf{g}_{d,k}^e$ учитывается при формировании опережающей составляющей, тогда как матрица обратной связи определяется для однородной части локальной модели.

Качество управления оценивалось квадратичным функционалом

$$J = \sum_{k=0}^{\infty} (\mathbf{x}_{e,k}^T Q \mathbf{x}_{e,k} + R \delta_k^2), \quad (2.74)$$

где матрица $Q \succeq 0$ задает относительную значимость ошибок состояния, а матрица $R \succ 0$ определяет штраф за величину управляющего воздействия. Увеличение элементов матрицы Q приводит к более жесткому подавлению соответствующих ошибок, тогда как увеличение элементов матрицы R уменьшает интенсивность рулевого воздействия и способствует более плавному управлению.

Минимизация функционала J для дискретной линейной системы приводит к закону управления с обратной связью по состоянию

$$\delta_{fb,k} = -K(\bar{v}_k) \mathbf{x}_{e,k}, \quad (2.75)$$

где матрица усиления $K(\bar{v}_k)$ определяется через решение дискретного алгебраического уравнения Риккати для замороженной пары $A_d^e(\bar{v}_k)$, B_d^e . Для фиксированного значения $\bar{v}$ уравнение имеет вид

$$\begin{aligned} P_L(\bar{v}) = & [A_d^e(\bar{v})]^T P_L(\bar{v}) A_d^e(\bar{v}) \\ & - [A_d^e(\bar{v})]^T P_L(\bar{v}) B_d^e \left(R + [B_d^e]^T P_L(\bar{v}) B_d^e \right)^{-1} \\ & \cdot [B_d^e]^T P_L(\bar{v}) A_d^e(\bar{v}) + Q. \end{aligned} \quad (2.76)$$

$$K(\bar{v}) = \left(R + [B_d^e]^T P_L(\bar{v}) B_d^e \right)^{-1} [B_d^e]^T P_L(\bar{v}) A_d^e(\bar{v}). \quad (2.77)$$

В вычислительном эксперименте профиль скорости постоянен в пределах каждого сценария, поэтому матрица $K(\bar{v})$ вычисляется один раз для соответствующей исследуемой скорости. При переменном профиле может применяться пересчет матрицы усиления либо выбор ее значения из заранее сформированной таблицы по текущей опорной скорости $\bar{v}_k$.

Одной лишь обратной связи по ошибкам недостаточно для точного прохождения криволинейных участков, поскольку при движении по дуге постоянной кривизны возможно появление установившейся ошибки. Для ее уменьшения используется опережающая составляющая. При $L = l_f + l_r$ коэффициент недостаточной поворачиваемости определяется как [65]

$$K_u = \frac{m}{L} \left(\frac{l_r}{C_f} - \frac{l_f}{C_r} \right), \quad (2.78)$$

а требуемый в установившемся режиме угол в линейном приближении имеет вид

$$\delta_{ff,k} = (L + K_u \bar{v}_k^2) \kappa_k^{\text{ref}}. \quad (2.79)$$

С учетом обратной связи закон управления записывается следующим образом:

$$\delta_k = \delta_{ff,k} + \delta_{fb,k}, \quad (2.80)$$

где $\delta_{ff,k}$ — угол поворота, необходимый для установившегося движения по траектории с кривизной κ_k^{ref} , а $\delta_{fb,k}$ — корректирующее воздействие, формируемое по текущим ошибкам состояния.

С учетом ограничений рулевого привода итоговое управляющее воздействие задавалось с насыщением по углу поворота:

$$\delta_k = \text{sat}_{[-\delta_{\max}, \delta_{\max}]} (\delta_{ff,k} - K(\bar{v}_k) \mathbf{x}_{e,k}), \quad (2.81)$$

а также с ограничением по скорости изменения угла поворота:

$$\delta_k \in [\delta_{k-1} - \dot{\delta}_{\max} T_s, \delta_{k-1} + \dot{\delta}_{\max} T_s]. \quad (2.82)$$

Построение регулятора в настоящей работе включало следующие этапы: получение линеаризованной модели ошибок бокового движения автомобиля; выбор вектора состояния, характеризующего отклонение от опорной траектории; дискретизацию модели с заданным шагом T_s ; задание матриц весовых коэффициентов Q и R ; решение дискретного алгебраического уравнения Риккати; вычисление матрицы усиления K и формирование закона управления; добавление опережающей составляющей по кривизне опорной траектории; введение ограничений рулевого привода.

На каждом шаге работы системы определялась ближайшая точка опорной траектории, вычислялись поперечная и курсовая ошибки, формировался вектор состояния $\mathbf{x}_{e,k}$, после чего по матрице усиления $K(\bar{v}_k)$ рассчитывалось корректирующее рулевое воздействие. Далее к нему добавлялась опережающая составляющая, связанная с кривизной опорной траектории, и результирующее значение подавалось на рулевой привод с учетом физических ограничений исполнительного механизма.

Таким образом, линейно-квадратичный регулятор был построен как регулятор состояния для линеаризованной модели бокового движения автомобиля. Его применение позволяет формализованно учитывать компромисс между точностью следования траектории и интенсивностью рулевого воздействия, что делает данный подход удобным для сопоставления с другими алгоритмами управления в одинаковых условиях вычислительного эксперимента.

2.5.3 Принцип работы алгоритма модельно-предиктивного управления (Model Predictive Control, MPC)

Модельно-предиктивное управление (MPC) относится к методам оптимального управления с убывающим (скользящим) горизонтом. На каждом такте управления решается задача оптимизации на конечном горизонте прогнозирования, формируется последовательность управляющих воздействий, после чего в объект управления подается только первое значение найденной последовательности. На следующем такте состояние измеряется повторно, горизонт прогнозирования сдвигается на один шаг, и задача оптимизации решается заново, что обеспечивает встроенную обратную связь и компенсацию возмущений.

Блок-схема алгоритма представлена на рисунке 13.

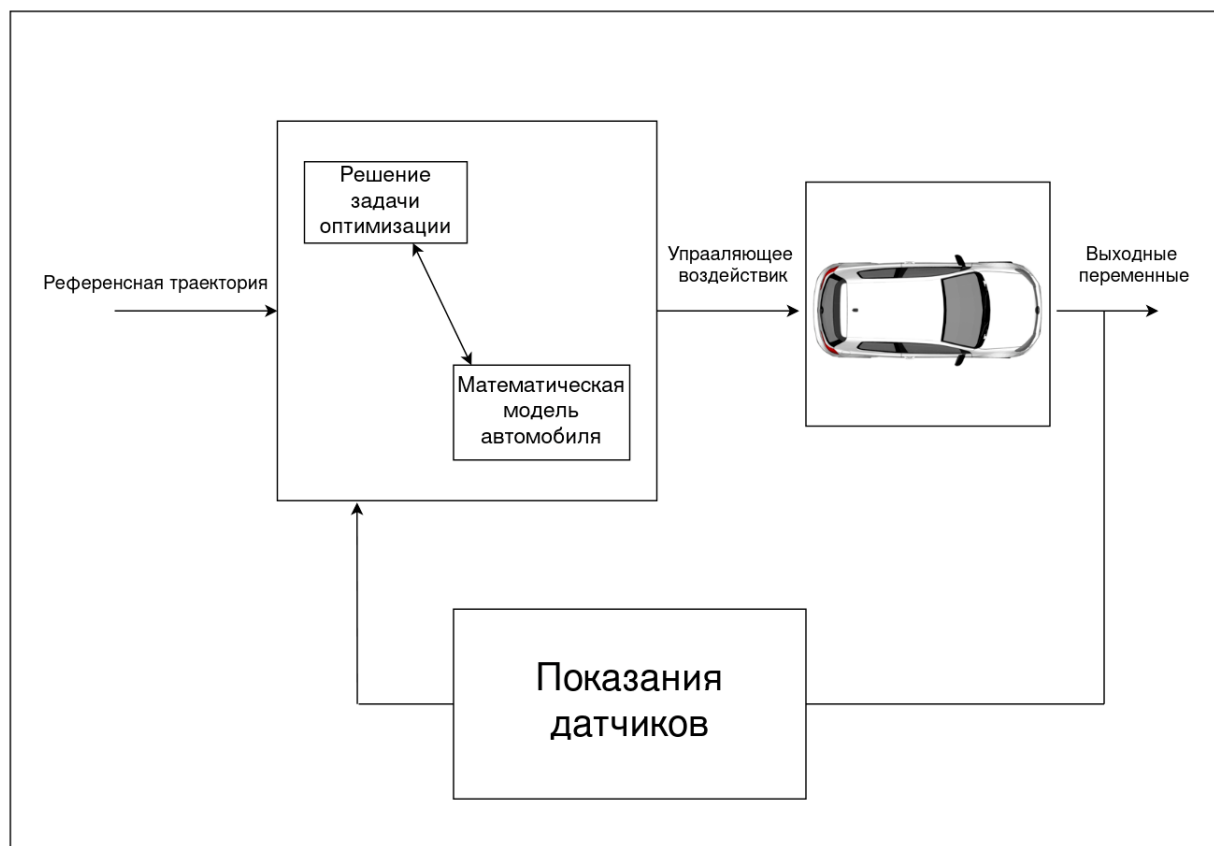


Рисунок 13 – Блок-схема построения алгоритма слежения за траекторией на основе MPC

В настоящей работе MPC используется как регулятор поперечного слежения за заранее сгенерированной опорной траекторией. Геометрические ограничения дорожного коридора учитываются на этапе планирования и формирования опорной траектории (см. главу 3). Продольный профиль также формируется до вызова поперечного регулятора. Выходом QR-решателя является только последовательность углов поворота передних колес, а изменение продольной скорости данным решателем не выполняется.

Измеряемое состояние транспортного средства задается в декартовой системе координат (x, y, ψ, v) . Вектор состояния MPC формируется по отношению к опорной точке (2.36) и в соответствии с (2.45) имеет вид

$$\mathbf{x}_{e,k} = \begin{bmatrix} e_{y,k} & \dot{e}_{y,k} & e_{\psi,k} & \dot{e}_{\psi,k} \end{bmatrix}^T. \quad (2.83)$$

Единственным управляющим воздействием является угол поворота управляемых колес δ_k . Для учета плавности управления используется его приращение

$$\Delta\delta_k = \delta_k - \delta_{k-1}. \quad (2.84)$$

В MPC используется та же модель поперечной динамики, что и при синтезе LQR:

$$\dot{\mathbf{x}}_e(t) = A_e(\bar{v}(t)) \mathbf{x}_e(t) + B_e \delta(t) + G_e(\bar{v}(t)) r^{\text{ref}}(t), \quad (2.85)$$

где матрицы $A_e(\bar{v})$, B_e и $G_e(\bar{v})$ определены выражениями (2.49) и (2.50). Таким образом, различие между физической моделью (2.42) и расчетной моделью MPC обусловлено только преобразованием координат (2.47). Матрица B_e в непрерывном времени от скорости не зависит, а G_e обозначает канал известного задающего воздействия и не является матрицей выхода.

Дискретизация по времени выполняется с шагом T_s и горизонтом прогнозирования N_p . В реализованной схеме используется $T_s = \Delta t$ и $N_p = 35$, что соответствует длительности прогнозирования $T_h = N_p T_s$. Выбор горизонта прогнозирования $N_p = 35$ обусловлен необходимостью обеспечить достаточную глубину предсказания при сохранении вычислительной реализуемости алгоритма в режиме реального времени. При принятом шаге дискретизации $T_s = 0.1$ с длительность горизонта составляет $T_h = N_p T_s = 3.5$ с. Такой горизонт позволяет учитывать не только текущее состояние объекта управления, но и предстоящие изменения геометрии опорной траектории на достаточно протяженном участке пути. Для исследованных постоянных опорных скоростей $\bar{v} = v^{\text{ref}} \in \{3, 5, 10, 13.88\}$ м/с это соответствует пространственному упреждению порядка $\bar{v} T_h$, то есть приблизительно 10.5, 17.5, 35.0 и 48.6 м соответственно. Следовательно, регулятор получает возможность заблаговременно формировать управляющее воздействие при входе на криволинейные участки траектории и при прохождении переходных зон.

С другой стороны, увеличение N_p приводит к росту размерности задачи квадратичного программирования, поскольку длина вектора оптимизируемых управлений равна N_p , а размерность стека прогнозируемых состояний составляет $4N_p$. При $N_p = 35$ указанные размерности остаются умеренными, что обеспечивает вычислительную эффективность решения. Результаты вычислительного эксперимента показывают, что при такой настройке среднее время расчета управляющего воздействия для MPC составляет около 2.4–2.7 мс, а 95%-квантиль не превышает 5.1 мс, что значительно меньше периода дискретизации $T_s = 100$ мс. Таким образом, значение $N_p = 35$ выбрано как рациональное: оно обеспечивает достаточную глубину прогноза для качественного поперечного слежения и одновременно сохраняет необходимый

запас по времени для реализации алгоритма в реальном времени.

На i -м интервале горизонта используются известные параметры $\bar{v}_{k+i} = v_{k+i}^{\text{ref}}$ и $r_{k+i}^{\text{ref}} = \bar{v}_{k+i} \kappa_{k+i}^{\text{ref}}$. При дискретизации явным методом Эйлера сначала формируются интервальные матрицы

$$\begin{aligned} A_{d,i}^e &= I + T_s A_e(\bar{v}_{k+i}), \\ B_{d,i}^e &= T_s B_e, \\ \mathbf{g}_{d,i}^e &= T_s G_e(\bar{v}_{k+i}) r_{k+i}^{\text{ref}}. \end{aligned} \quad (2.86)$$

Поскольку определение $\mathbf{x}_e$ зависит от $\bar{v}$ и r^{ref} , при изменении опорных параметров между соседними интервалами состояние в конце интервала должно быть выражено в координатах следующей опорной точки. Для этого вводятся

$$\mathcal{R}_i = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & \bar{v}_{k+i+1} - \bar{v}_{k+i} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \boldsymbol{\eta}_i = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -(r_{k+i+1}^{\text{ref}} - r_{k+i}^{\text{ref}}) \end{bmatrix}. \quad (2.87)$$

Эффективные матрицы линейной нестационарной модели определяются выражениями

$$A_{d,i} = \mathcal{R}_i A_{d,i}^e, \quad B_{d,i} = \mathcal{R}_i B_{d,i}^e, \quad \mathbf{g}_{d,i} = \mathcal{R}_i \mathbf{g}_{d,i}^e + \boldsymbol{\eta}_i, \quad (2.88)$$

а дискретная модель принимает вид

$$\mathbf{x}_{e,k+i+1|k} = A_{d,i} \mathbf{x}_{e,k+i|k} + B_{d,i} \delta_{k+i|k} + \mathbf{g}_{d,i}, \quad i = 0, \dots, N_p - 1. \quad (2.89)$$

Для вычисления последнего преобразования в (2.87) на вход регулятора передаются $N_p + 1$ последовательных опорных точек. При постоянных в пределах горизонта $\bar{v}$ и r^{ref} выполняются $\mathcal{R}_i = I$ и $\boldsymbol{\eta}_i = 0$, поэтому (2.88) сводится непосредственно к (2.86). В общем случае матрицы $A_{d,i}$, $B_{d,i}$ и вектор $\mathbf{g}_{d,i}$ различаются между интервалами и пересчитываются на каждом такте управления по известной последовательности опорных скоростей и кривизн.

Целевая функция MPC формируется как квадратичный функционал, минимизирующий ошибки слежения, величину угла поворота колес и его

приращения:

$$J = \sum_{i=1}^{N_p} \mathbf{x}_{e,k+i|k}^T Q \mathbf{x}_{e,k+i|k} + \sum_{i=0}^{N_p-1} \left(R \delta_{k+i|k}^2 + R_\Delta \Delta \delta_{k+i|k}^2 \right), \quad (2.90)$$

где $Q \succeq 0$, $R \succ 0$, $R_\Delta \succeq 0$ — весовые матрицы. В вычислительных экспериментах используются диагональные матрицы, согласованные с настройками регулятора поперечного слежения, при этом штрафование $\Delta \delta$ обеспечивает сглаживание управляющего воздействия.

При постановке задачи MPC используются ограничения, обеспечивающие физическую реализуемость управления рулевым приводом. Ограничение по углу поворота управляемых колес имеет вид

$$-\delta_{\max} \leq \delta_{k+i|k} \leq \delta_{\max}, \quad i = 0, \dots, N_p - 1. \quad (2.91)$$

Ограничение по скорости изменения угла поворота задается через приращения

$$-\Delta \delta_{\max} \leq \Delta \delta_{k+i|k} \leq \Delta \delta_{\max}, \quad \Delta \delta_{\max} = \dot{\delta}_{\max} T_s. \quad (2.92)$$

Дополнительно вводится консервативная квазистационарная граница по поперечному ускорению. С учетом кинематического соотношения

$$a_y \approx \bar{v}_{k+i}^2 \kappa, \quad (2.93)$$

и приближенной связи $\kappa \approx \tan(\delta)/L$ ограничение $|a_y| \leq a_{y,\max}$ приводит к следующей суррогатной границе допустимого угла поворота колес:

$$|\delta_{k+i|k}| \leq \arctan\left(\frac{a_{y,\max} L}{\bar{v}_{k+i}^2}\right), \quad i = 0, \dots, N_p - 1. \quad (2.94)$$

Данное ограничение используется совместно с (2.91) и формирует зависящую от известной опорной скорости допустимую область δ на горизонте. Оно не заменяет вычисление поперечного ускорения по динамической модели в предельных режимах.

С учетом модели (2.89), функционала (2.90) и ограничений (2.91)–(2.94) задача MPC приводится к задаче квадратичного программирования (QP) с линейными ограничениями. Это обеспечивает возможность решения задачи в

реальном времени с применением специализированных численных решателей, что используется далее в подразделе 2.5.4.

2.5.4 Преобразование задачи МРС к задаче квадратичного программирования (QP)

Для обеспечения вычислительной реализуемости алгоритма в режиме реального времени задача МРС в настоящей работе приводится к задаче квадратичного программирования (QP) с линейными ограничениями. Известные параметры опорной траектории на текущем такте объединяются в набор

$$\mathcal{P}_k = \left\{ \bar{v}_{k+i}, \kappa_{k+i}^{\text{ref}}, r_{k+i}^{\text{ref}} \right\}_{i=0}^{N_p}. \quad (2.95)$$

Дополнительная опорная точка с индексом $k + N_p$ необходима только для формирования конечного состояния и преобразования (2.87). В качестве переменной оптимизации рассматривается последовательность углов поворота управляемых колес на N_p интервалах:

$$\mathbf{U}_k = \left[\delta_{k|k} \quad \delta_{k+1|k} \quad \cdots \quad \delta_{k+N_p-1|k} \right]^T. \quad (2.96)$$

Следовательно, ни продольная скорость, ни кривизна, ни требуемая скорость рыскания в вектор решения QP не входят.

Для записи прогноза состояния вводится стек векторов состояний (ошибок слежения) на горизонте:

$$\mathbf{X}_k = \left[\mathbf{x}_{e,k+1|k}^T \quad \mathbf{x}_{e,k+2|k}^T \quad \cdots \quad \mathbf{x}_{e,k+N_p|k}^T \right]^T, \quad (2.97)$$

где $\mathbf{x}_{e,k} \in \mathbb{R}^{n_x}$, $n_x = 4$ (см. (2.83)), а $\delta_k \in \mathbb{R}$ является скалярным управлением.

Для компактности далее индекс k в интервальных матрицах опускается. Дискретная модель ошибок на каждом шаге горизонта имеет вид

$$\mathbf{x}_{e,k+i+1|k} = A_{d,i} \mathbf{x}_{e,k+i|k} + B_{d,i} \delta_{k+i|k} + \mathbf{g}_{d,i}, \quad i = 0, \dots, N_p - 1, \quad (2.98)$$

где все коэффициенты вычисляются до вызова решателя по формулам (2.86)–(2.88).

Для линейной нестационарной модели вводится матрица перехода

$$\Phi(i, j) = \begin{cases} A_{d,i-1}A_{d,i-2} \cdots A_{d,j}, & i > j, \\ I, & i = j. \end{cases} \quad (2.99)$$

Тогда состояние на i -м шаге прогнозирования определяется выражением

$$\begin{aligned} \mathbf{x}_{e,k+i|k} &= \Phi(i, 0)\mathbf{x}_{e,k} + \sum_{j=0}^{i-1} \Phi(i, j+1)B_{d,j}\delta_{k+j|k} \\ &+ \sum_{j=0}^{i-1} \Phi(i, j+1)\mathbf{g}_{d,j}, \quad i = 1, \dots, N_p. \end{aligned} \quad (2.100)$$

В отличие от стационарного случая, в (2.99) используются упорядоченные произведения различных матриц, а не степени одной матрицы A_d .

После объединения прогнозируемых состояний выражение (2.100) принимает вид

$$\mathbf{X}_k = \mathcal{A}_k \mathbf{x}_{e,k} + \mathcal{B}_k \mathbf{U}_k + \mathcal{G}_k, \quad (2.101)$$

где $\mathcal{A}_k \in \mathbb{R}^{(N_p n_x) \times n_x}$, $\mathcal{B}_k \in \mathbb{R}^{(N_p n_x) \times N_p}$ и $\mathcal{G}_k \in \mathbb{R}^{N_p n_x}$ формируются из последовательностей $A_{d,i}$, $B_{d,i}$ и $\mathbf{g}_{d,i}$. Эти величины пересчитываются на каждом такте управления по набору $\mathcal{P}_k$. Для удобства вводится аффинная составляющая прогноза

$$\mathbf{X}_{\text{aff},k} = \mathcal{A}_k \mathbf{x}_{e,k} + \mathcal{G}_k, \quad \mathbf{X}_k = \mathbf{X}_{\text{aff},k} + \mathcal{B}_k \mathbf{U}_k. \quad (2.102)$$

Аффинная составляющая прогноза определяется текущим состоянием и известными параметрами опорной траектории и не зависит от оптимизируемого вектора управляющих воздействий.

Поскольку $\mathbf{x}_e$ является вектором ошибок слежения, целевое состояние в принятой малосигнальной модели соответствует $\mathbf{x}_e = 0$. Тогда функционал (2.90) записывается в компактной форме:

$$J = \mathbf{X}_k^\top \bar{Q} \mathbf{X}_k + \mathbf{U}_k^\top \bar{R} \mathbf{U}_k + (\mathbf{E} \mathbf{U}_k - \mathbf{d}_{\delta,k})^\top \bar{R}_\Delta (\mathbf{E} \mathbf{U}_k - \mathbf{d}_{\delta,k}), \quad (2.103)$$

где $\bar{Q} = \text{blkdiag}(Q, \dots, Q) \in \mathbb{R}^{(N_p n_x) \times (N_p n_x)}$, $\bar{R} = \text{blkdiag}(R, \dots, R) \in \mathbb{R}^{N_p \times N_p}$, $\bar{R}_\Delta = \text{blkdiag}(R_\Delta, \dots, R_\Delta) \in \mathbb{R}^{N_p \times N_p}$. Матрица разностей $\mathbf{E} \in \mathbb{R}^{N_p \times N_p}$ реализует

оператор приращений $\Delta\delta$:

$$\Delta\mathbf{U}_k = \begin{bmatrix} \delta_{k|k} - \delta_{k-1} \\ \delta_{k+1|k} - \delta_{k|k} \\ \vdots \\ \delta_{k+N_p-1|k} - \delta_{k+N_p-2|k} \end{bmatrix} = \mathbf{E}\mathbf{U}_k - \mathbf{d}_{\delta,k}, \quad (2.104)$$

где

$$\mathbf{E} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ -1 & 1 & 0 & \cdots & 0 \\ 0 & -1 & 1 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & -1 & 1 \end{bmatrix}, \quad \mathbf{d}_{\delta,k} = [\delta_{k-1} \ 0 \ \cdots \ 0]^\top. \quad (2.105)$$

Подставляя (2.102) в (2.103) и группируя члены по $\mathbf{U}_k$, получаем квадратичную задачу вида, принятого в решателе OSQP:

$$\min_{\mathbf{U}_k} \frac{1}{2} \mathbf{U}_k^\top P_k \mathbf{U}_k + \mathbf{q}_k^\top \mathbf{U}_k, \quad (2.106)$$

где матрица $P_k \succeq 0$ и вектор $\mathbf{q}_k$ определяются следующим образом:

$$P_k = 2 \left(\mathcal{B}_k^\top \bar{Q} \mathcal{B}_k + \bar{R} + \mathbf{E}^\top \bar{R}_\Delta \mathbf{E} \right), \quad (2.107)$$

$$\mathbf{q}_k = 2 \left(\mathcal{B}_k^\top \bar{Q} \mathbf{X}_{\text{aff},k} - \mathbf{E}^\top \bar{R}_\Delta \mathbf{d}_{\delta,k} \right). \quad (2.108)$$

Постоянные по $\mathbf{U}_k$ слагаемые в функционале J на оптимум не влияют и в (2.106) не включаются.

Ограничения на управление формируются в виде линейных двусторонних неравенств $\mathbf{l}_k \leq A_{\text{qp}} \mathbf{U}_k \leq \mathbf{u}_k$, где матрица ограничений строится путем вертикального объединения ограничений на угол поворота и на его приращение:

$$A_{\text{qp}} = \begin{bmatrix} I \\ \mathbf{E} \end{bmatrix}, \quad (2.109)$$

а векторы нижних и верхних границ имеют вид

$$\mathbf{l}_k = \begin{bmatrix} \delta_{\min} \\ -\Delta\delta_{\max} \mathbf{1} + \mathbf{d}_{\delta,k} \end{bmatrix}, \quad \mathbf{u}_k = \begin{bmatrix} \delta_{\max} \\ \Delta\delta_{\max} \mathbf{1} + \mathbf{d}_{\delta,k} \end{bmatrix}. \quad (2.110)$$

Здесь $\delta_{\min} \in \mathbb{R}^{N_p}$, $\delta_{\max} \in \mathbb{R}^{N_p}$ — покомпонентные границы для $\delta_{k+i|k}$. В частности, с учетом ограничений рулевого привода и ограничения по поперечному ускорению (2.94) верхняя граница на i -м шаге горизонта задается как

$$\delta_{\max,k+i} = \min\left(\delta_{\max}, \arctan\left(\frac{a_{y,\max}L}{\bar{v}_{k+i}^2}\right)\right), \quad \delta_{\min,k+i} = -\delta_{\max,k+i}. \quad (2.111)$$

Параметр $\Delta\delta_{\max} = \dot{\delta}_{\max}T_s$ соответствует ограничению на скорость изменения угла поворота (см. (2.92)), $\mathbf{1}$ — вектор единиц размерности N_p .

Таким образом, задача управления на каждом такте сводится к решению QP (2.106) при ограничениях (2.109)–(2.110). Полученная оптимальная последовательность $\mathbf{U}_k^*$ используется по принципу убывающего горизонта: в объект управления подается только первое значение $\delta_{k|k}^*$, после чего задача формируется повторно с обновленными $\mathbf{x}_{e,k}$, набором $\mathcal{P}_k$ и предыдущим углом δ_{k-1} .

2.5.5 Результаты моделирования и воспроизводимость вычислительного эксперимента

Цель вычислительного эксперимента состоит в сопоставлении качества поперечного слежения и вычислительных затрат для трех алгоритмов управления: алгоритма геометрического преследования (PP), линейно-квадратичного регулятора (LQR) и модельно-предиктивного управления (MPC). Для устранения методической неоднозначности далее отдельно рассматриваются модель объекта управления, применяемая для пересчета состояния транспортного средства при моделировании, и расчетные модели, используемые непосредственно в алгоритмах управления.

Исходные данные и условия вычислительного эксперимента. Опорная траектория задается дискретным набором точек $\{P_i^{\text{ref}}\}$ вида (2.36), содержащим как прямолинейные, так и криволинейные участки. Шаг дискретизации при моделировании и в алгоритмах управления принят равным $T_s = 0.1$ с. Для оценки влияния скорости движения рассматриваются четыре режима: $\bar{v} = v^{\text{ref}} \in \{3, 5, 10, 13.88\}$ м/с. При этом геометрия опорной траектории во всех сериях расчетов сохраняется неизменной. Начальное состояние принимается совпадающим с опорной траекторией, то есть начальные ошибки по положению и курсу равны нулю. Это позволяет исключить влияние искусственно задаваемого начального рассогласования на результаты сопоставления алгоритмов.

Построение опорной траектории для тестирования контроллеров. Для воспроизводимости сравнения PP, LQR и MPC опорная траектория формировалась одинаковым способом во всех сериях эксперимента:

1. задавался набор контрольных точек S-образной трассы (таблица 2);
2. по контрольным точкам строилась гладкая кривая кубическим B-сплайном;
3. кривая параметризовалась длиной дуги и равномерно пересэмплировалась с шагом $\Delta s = 0.5$ м;
4. для каждой дискретной точки вычислялись ψ_i^{ref} и κ_i^{ref} , после чего задавался профиль v_i^{ref} (постоянный в рамках каждого сценария скорости).

Пространственная сетка с шагом Δs используется как таблица для интерполяции, а не как временная сетка МРС. Если s_k — дуговая координата ближайшей к автомобилю точки, то параметры i -й опорной точки горизонта выбираются при

$$s_{k+i|k} = s_k + T_s \sum_{j=0}^{i-1} \bar{v}_{k+j}, \quad i = 1, \dots, N_p, \quad (2.112)$$

после чего X^{ref} , Y^{ref} , ψ^{ref} , κ^{ref} и v^{ref} интерполируются по дуговой координате. Тем самым пространственная опорная траектория согласуется с временным шагом прогнозной модели при произвольном известном профиле скорости. Следовательно, данный эксперимент характеризует влияние уровня постоянной скорости, но не предназначен для оценки переходных процессов при изменении $\bar{v}_{k+i}$ внутри горизонта. Линейная нестационарная постановка (2.89) сохраняется для переменного профиля, однако такой режим требует отдельной экспериментальной проверки.

Таблица 2 – Контрольные точки S-образной опорной траектории в вычислительном эксперименте

i	$x_i, \text{ м}$	$y_i, \text{ м}$
0	0	0.0
1	20	0.0
2	40	3.0
3	60	5.5
4	80	0.0
5	100	-5.5
6	120	-3.0
7	140	0.0

Курсовой угол и кривизна опорной траектории определялись по дискретным производным:

$$\psi_i^{\text{ref}} = \text{atan2}(Y_{i+1}^{\text{ref}} - Y_i^{\text{ref}}, X_{i+1}^{\text{ref}} - X_i^{\text{ref}}), \quad \kappa_i^{\text{ref}} = \frac{\psi_{i+1}^{\text{ref}} - \psi_i^{\text{ref}}}{s_{i+1} - s_i}. \quad (2.113)$$

Таким образом, все контроллеры тестировались на идентичной опорной геометрии и различались только законом формирования управляющего воздействия.

Модель объекта управления в вычислительном эксперименте. Пересчет состояния транспортного средства при моделировании выполняется на основе кинематической велосипедной модели в декартовой системе координат (x, y, ψ, v) :

$$\begin{aligned}x_{k+1} &= x_k + v_k \cos \psi_k T_s, \\y_{k+1} &= y_k + v_k \sin \psi_k T_s, \\ \psi_{k+1} &= \psi_k + \frac{v_k}{L} \tan(\delta_k) T_s, \\v_{k+1} &= v_k + a_k T_s,\end{aligned}\tag{2.114}$$

где L — колесная база, δ_k — угол поворота управляемых колес, a_k — продольное ускорение. Продольное движение стабилизируется простым регулятором скорости:

$$a_k = \text{sat}_{[a_{\min}, a_{\max}]}(K_v (\bar{v}_k - v_k)),\tag{2.115}$$

где K_v — коэффициент усиления, $a_{\min}$ и $a_{\max}$ — ограничения по продольному ускорению.

Расчетные модели, используемые в алгоритмах управления. Алгоритмы PP, LQR и MPC используют измеряемое состояние (x, y, ψ, v) для вычисления отклонений относительно ближайшей точки опорной траектории $P^{\text{ref}} = (X^{\text{ref}}, Y^{\text{ref}}, \psi^{\text{ref}}, \kappa^{\text{ref}}, v^{\text{ref}})$. В качестве основных переменных ошибки рассматриваются поперечное отклонение e_y , ошибка по курсу e_ψ и их производные. При построении матриц LQR и MPC принимается $\bar{v} = v^{\text{ref}}$, тогда как измеренная скорость v используется для оценки состояния и в отдельном продольном контуре (2.115). Таким образом, продольная скорость не входит в вектор решений поперечного QR. Прогнозирование и синтез поперечного управления выполняются на основе линеаризованной динамической модели в координатах ошибок, тогда как модель объекта управления в вычислительном эксперименте остается кинематической в соответствии с (2.114).

Поскольку состояние модели объекта (2.114) не содержит непосредственно измеряемых v_y и r , производные ошибок в вычислительном эксперименте

оцениваются конечными разностями:

$$\dot{e}_{y,k} = \frac{e_{y,k} - e_{y,k-1}}{T_s}, \quad \dot{e}_{\psi,k} = \frac{\text{wrap}_{(-\pi,\pi]}(e_{\psi,k} - e_{\psi,k-1})}{T_s}. \quad (2.116)$$

На первом шаге принимается $\dot{e}_{y,0} = \dot{e}_{\psi,0} = 0$. В натурной системе вместо (2.116) могут использоваться оценки v_y и r , получаемые наблюдателем состояния, с последующим применением соотношений (2.40).

Параметры транспортного средства и ограничения. В таблице 3 приведены параметры модели транспортного средства, использованные при моделировании, а также при построении расчетных моделей алгоритмов LQR и МРС. Ограничения рулевого управления задаются предельными значениями угла поворота $\delta_{\max}$ и скорости его изменения $\dot{\delta}_{\max}$.

Таблица 3 – Параметры моделируемого транспортного средства

Условное обозначение	Наименование параметра	Значение
m	Полная масса транспортного средства	1559 [кг]
I_z	Момент инерции относительно вертикальной оси	2900 [кг · м ²]
$\bar{v} = v^{\text{ref}}$	Опорная скорость в сериях вычислительного эксперимента	3, 5, 10, 13.88 [м/с]
$v_{\min}$	Нижняя граница применимости динамической поперечной модели	3 [м/с]
l_f	Расстояние от центра масс до передней оси	1.14 [м]
l_r	Расстояние от центра масс до задней оси	1.495 [м]
C_f	Коэффициент сопротивления боковому уводу передней оси	120000 [Н/рад]
C_r	Коэффициент сопротивления боковому уводу задней оси	120000 [Н/рад]

Указанные значения коэффициентов сопротивления боковому уводу заимствованы из работы П.А. Васина [11], в которой они использовались при параметризации математической модели движения автомобиля «Лада Веста», имеющего аналогичные массово-геометрические характеристики.

Параметры алгоритмов управления. С целью обеспечения воспроизводимости вычислительного эксперимента в таблице 4 приведены используемые настройки алгоритмов PP, LQR и MPC. Для алгоритма PP применяется правило выбора дальности упреждения $l_d(v)$ согласно (2.66); в частном случае допускается использование постоянного значения $l_d = \text{const}$. Для LQR указываются весовые матрицы Q и R . Для MPC фиксируются горизонт прогнозирования N_p , шаг дискретизации T_s , весовые коэффициенты функционала качества и активные ограничения.

Таблица 4 – Параметры алгоритмов управления, использованные при моделировании

Алгоритм	Настройки
PP	$l_d(v) = \text{sat}_{[l_{\min}, l_{\max}]}(l_0 + k_v v)$; δ определяется по (2.65) с насыщением (2.67); ограничение на скорость изменения угла задается по (2.68)
LQR	$\mathbf{x}_e = [e_y, \dot{e}_y, e_\psi, \dot{e}_\psi]^T$; $Q = \text{diag}(2.0, 0.5, 1.5, 0.2)$; $R = [1.0]$; $K(\bar{v})$ вычисляется для скорости соответствующего сценария
MPC	$\mathbf{x}_e = [e_y, \dot{e}_y, e_\psi, \dot{e}_\psi]^T$; $N_p = 35$, $T_s = 0.1$ с; $Q = \text{diag}(6.0, 1.2, 8.0, 0.8)$, $R = [1.0]$, $R_\Delta = [0.4]$; дискретизация по Эйлера; $\bar{v}$ — известный параметр; ограничения (2.91)–(2.94)

Показатели качества и вычислительные затраты. Качество слежения оценивается по поперечной ошибке e_y и, при необходимости, по продольной ошибке e_s , вычисляемым относительно ближайшей точки опорной траектории:

$$\begin{aligned} e_y &= -\sin(\psi_{\text{ref}})(x - x_{\text{ref}}) + \cos(\psi_{\text{ref}})(y - y_{\text{ref}}), \\ e_s &= -\cos(\psi_{\text{ref}})(x - x_{\text{ref}}) - \sin(\psi_{\text{ref}})(y - y_{\text{ref}}). \end{aligned} \quad (2.117)$$

Для каждой метрики $z \in \{|e_y|, |e_s|\}$ определяются среднеквадратичная ошибка, максимальное значение и 95%-квантиль. Для управляющего воздействия дополнительно оцениваются величины $\max |\delta|$ и $\text{STD}(\delta)$. Вычислительные затраты характеризуются временем формирования управляющего воздействия на одном такте:

$$t_k = (t_k^{\text{end}} - t_k^{\text{start}}) \cdot 10^3 \text{ [мс]}, \quad (2.118)$$

после чего определяются среднее значение $\bar{t}$ и 95%-квантиль $q_{0.95}(t)$. Измерение времени выполняется с использованием высокоточного монотонного таймера `perf_counter`. В указанную величину включается только вычисление управляющего воздействия, в том числе решение задачи квадратичного программирования для MPC; пересчет состояния объекта управления по (2.114) и графическое отображение результатов в эту оценку не включаются.

Вычислительная платформа. Для обеспечения воспроизводимости результатов должны быть указаны основные характеристики вычислительной платформы: модель процессора, объем оперативной памяти, операционная система, версия среды выполнения программ, а также версии используемых библиотек численной оптимизации. Рекомендуемый формат представления таких сведений приведен в таблице 5.

Таблица 5 – Характеристики вычислительной платформы

Параметр	Значение
Процессор	Intel Core i5-10400F
Оперативная память	32
Операционная система	Ubuntu 24.04
Среда выполнения программ	Python 3.12

Сводные результаты. Итоговые выводы формируются на основе единой сводной таблицы метрик по скоростям. Таблицы 6 и 7, а также рисунки 14-17 содержат результаты для $\bar{v} \in \{3, 5, 10, 13.88\}$ м/с.

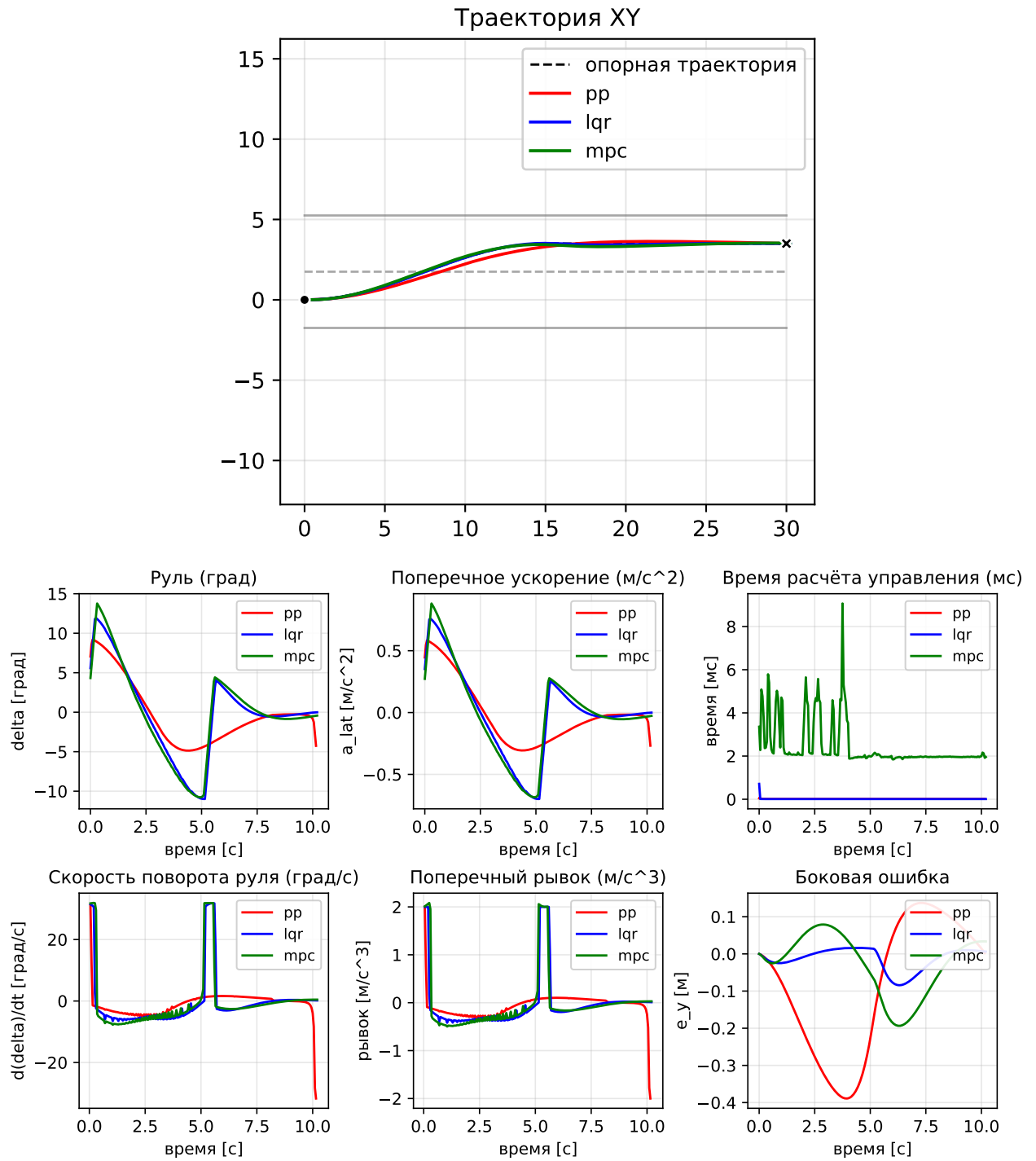
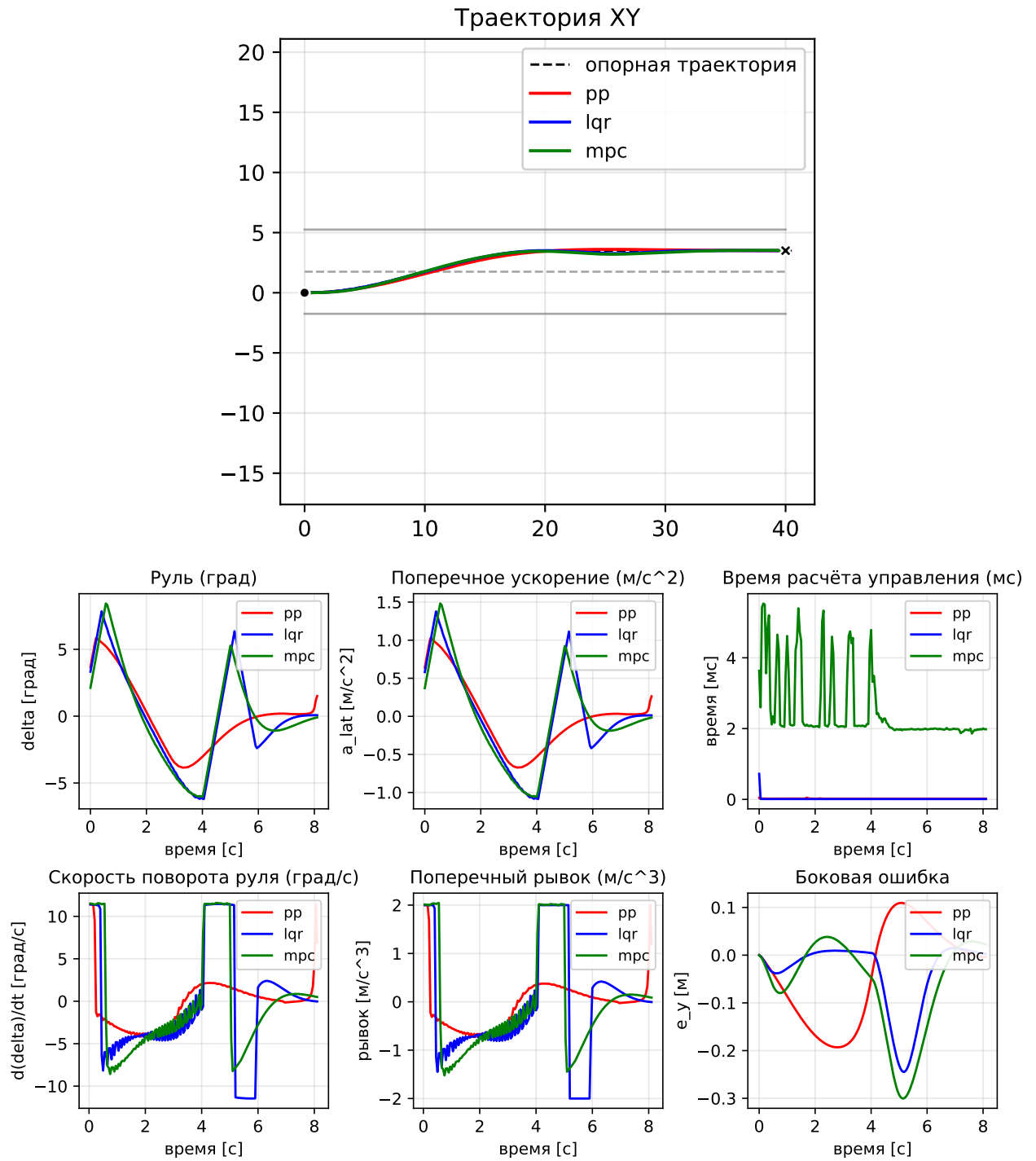


Рисунок 14 – Сравнение алгоритмов PP, LQR и MPC при скорости 3 м/с



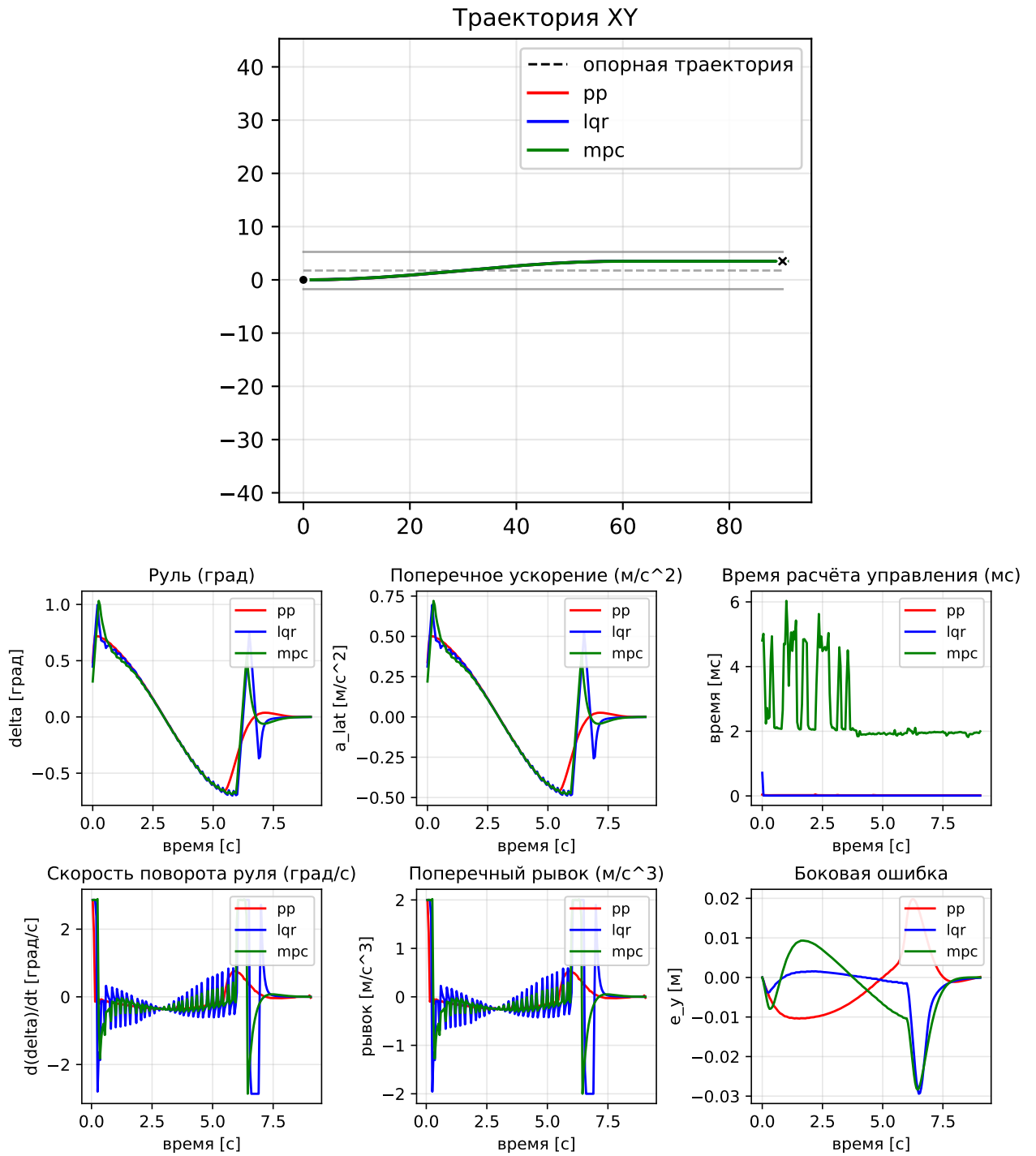


Рисунок 16 – Сравнение алгоритмов PP, LQR и MPC при скорости 10 м/с

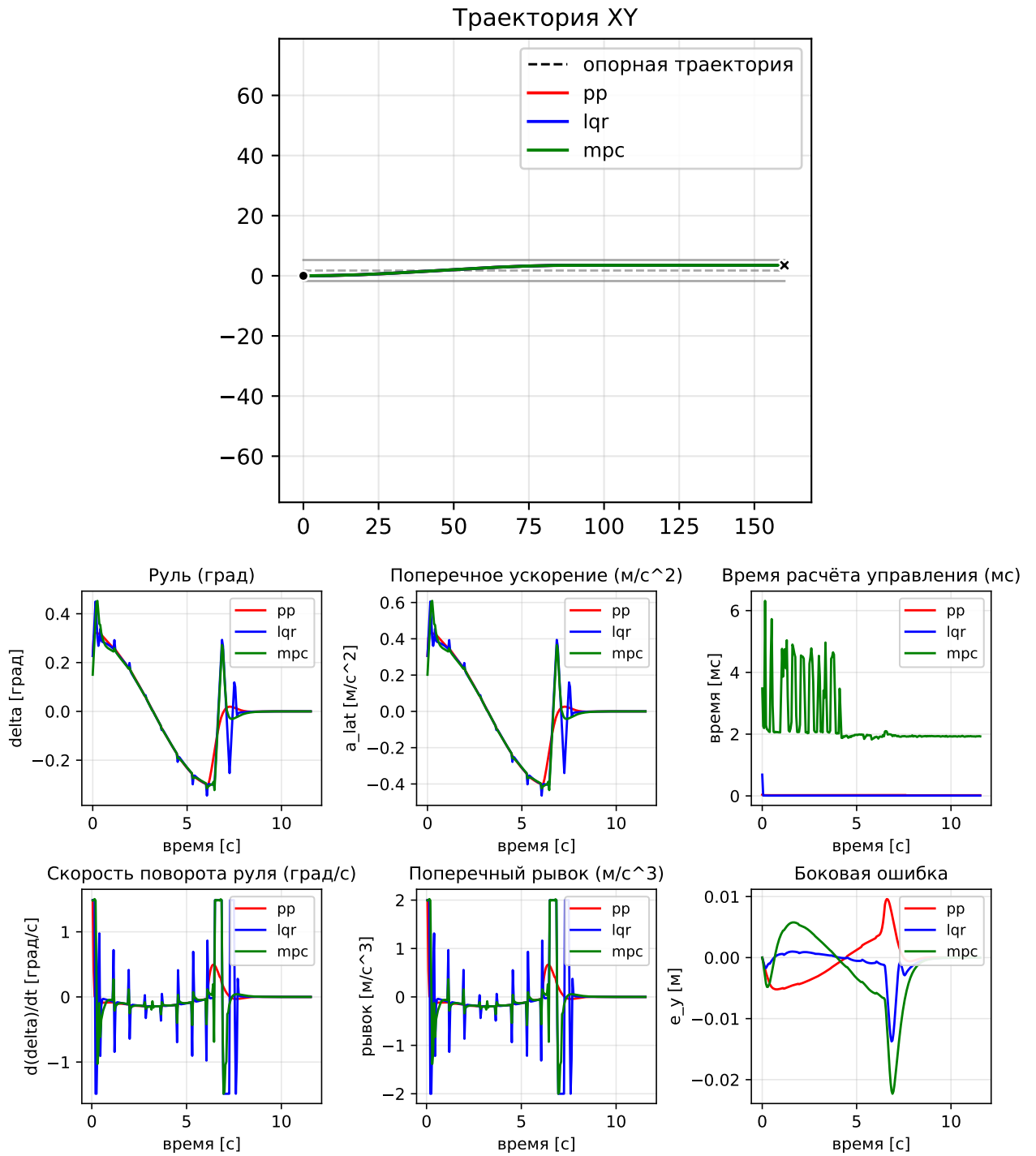


Рисунок 17 – Сравнение алгоритмов PP, LQR и MPC при скорости 13.88 м/с

Таблица 6 – Сводные метрики слежения и вычислительной сложности

$\bar{v}$, м/с	Регулятор	$\text{RMSE}(e_y)$, м	$\max e_y $, м	$q_{0.95}(e_y)$, м	$\text{RMSE}(e_s)$, м	$\max e_s $, м	$q_{0.95}(e_s)$, м	$\max \delta $, град	$\text{STD}(\delta)$, град
3.00	PP	0.199	0.389	0.382	0.151	0.173	0.160	9.215	3.943
3.00	LQR	0.032	0.084	0.079	0.150	0.244	0.160	11.929	5.541
3.00	MPC	0.086	0.194	0.188	0.150	0.244	0.160	13.767	5.792
5.00	PP	0.108	0.193	0.191	0.252	0.395	0.263	5.845	2.568
5.00	LQR	0.081	0.245	0.221	0.250	0.356	0.262	7.846	3.556
5.00	MPC	0.116	0.300	0.285	0.251	0.351	0.264	8.431	3.575
10.00	PP	0.015	0.034	0.028	0.503	0.831	0.529	1.367	0.592
10.00	LQR	0.038	0.125	0.178	0.503	0.783	0.529	3.602	1.263
10.00	MPC	0.038	0.122	0.105	0.503	0.830	0.529	2.003	0.686
13.88	PP	0.003	0.010	0.005	0.696	0.924	0.746	0.322	0.160
13.88	LQR	0.002	0.014	0.003	0.696	0.924	0.746	0.450	0.170
13.88	MPC	0.006	0.022	0.013	0.696	0.924	0.746	0.453	0.168

Таблица 7 – Вычислительные затраты регуляторов: среднее и 95%-квантиль времени расчета управления

$\bar{v}$, м/с	Регулятор	$\bar{t}$, мс	$q_{0.95}(t)$, мс
3.00	PP	0.021	0.022
3.00	LQR	0.030	0.010
3.00	MPC	2.394	4.590
5.00	PP	0.021	0.022
5.00	LQR	0.052	0.010
5.00	MPC	2.703	4.976
10.00	PP	0.021	0.022
10.00	LQR	0.040	0.011
10.00	MPC	2.579	5.053
13.88	PP	0.021	0.022
13.88	LQR	0.026	0.023
13.88	MPC	2.386	4.712

Превышение среднего времени вычисления над 95%-квантилем в отдельных режимах LQR не является противоречием: среднее значение чувствительно к единичным длительным вычислительным выбросам, тогда как $q_{0.95}(t)$ характеризует верхнюю границу времени для 95% зарегистрированных тактов управления.

Были заданы: параметры регуляторов, модели объекта управления, формула (2.114), внутренних моделей регуляторов, набора метрик, формула (2.117) и методики измерения времени, формула (2.118).

Анализ результатов

Результаты вычислительных экспериментов на S-образной трассе при $\bar{v} \in \{3, 5, 10, 13.88\}$ м/с позволяют сопоставить регуляторы PP, LQR и MPC по точности поперечного слежения, характеру управляющего воздействия и вычислительным затратам (рис. 14–17, табл. 6, 7).

По точности поперечного слежения не выявляется регулятор, обеспечивающий наилучший результат на всех исследованных скоростях. При $\bar{v} = 3$ м/с наименьшее значение $\text{RMSE}(e_y) = 0.032$ м получено для LQR, тогда

как для MPC и PP оно составляет соответственно 0.086 м и 0.199 м. При $\bar{v} = 5$ м/с наименьшее значение $\text{RMSE}(e_y) = 0.081$ м также обеспечивается LQR, для PP получено 0.108 м, для MPC — 0.116 м. При $\bar{v} = 10$ м/с наилучший результат показывает PP: $\text{RMSE}(e_y) = 0.015$ м; для MPC получено 0.038 м, для LQR — также 0.038 м. При $\bar{v} = 13.88$ м/с различия между алгоритмами минимальны; наименьшее значение $\text{RMSE}(e_y) = 0.002$ м получено для LQR, для PP — 0.003 м, для MPC — 0.006 м. Следовательно, LQR обеспечивает наименьшую среднеквадратическую поперечную ошибку в трех из четырех исследованных режимов, PP показывает лучший результат при $\bar{v} = 10$ м/с, а MPC сохраняет приемлемую точность слежения на всех скоростях. При $\bar{v} = 10$ м/с MPC и LQR совпадают по среднеквадратической поперечной ошибке.

Анализ 95%-квантиля поперечной ошибки $q_{0.95}(|e_y|)$ подтверждает указанную закономерность. При $\bar{v} = 3$ м/с наименьшее значение $q_{0.95}(|e_y|) = 0.079$ м получено для LQR, при $\bar{v} = 5$ м/с — для PP ($q_{0.95}(|e_y|) = 0.191$ м), при $\bar{v} = 10$ м/с — также для PP (0.028 м), при $\bar{v} = 13.88$ м/с — для LQR (0.003 м). MPC по данной метрике не занимает первого места, но при $\bar{v} = 10$ м/с показывает значительно лучший результат, чем LQR (0.105 м против 0.178 м). Полученные данные свидетельствуют о том, что в рассматриваемой постановке MPC не обеспечивает минимальную поперечную ошибку, но и не демонстрирует критического ухудшения качества слежения относительно лучших результатов на трассе.

По продольной ошибке e_s различия между алгоритмами выражены слабо. Для каждой фиксированной скорости значения $\text{RMSE}(e_s)$ и $q_{0.95}(|e_s|)$ практически совпадают для всех трех регуляторов: при $\bar{v} = 3$ м/с $\text{RMSE}(e_s) \approx 0.150$ м, при $\bar{v} = 5$ м/с — около 0.25 м, при $\bar{v} = 10$ м/с — 0.503 м, при $\bar{v} = 13.88$ м/с — 0.696 м. Это указывает на то, что в данной серии экспериментов различия между регуляторами определяются прежде всего качеством поперечного управления, тогда как продольная ошибка в большей степени задается опорным профилем скорости и принятой процедурой вычисления метрик.

Характер управляющего воздействия различается более отчетливо. По максимальному углу поворота рулевого колеса $\max |\delta|$ на всех исследованных скоростях наименьшие значения обеспечивает PP: 9.215° при 3 м/с, 5.845° при 5 м/с, 1.367° при 10 м/с и 0.322° при 13.88 м/с. Для LQR и MPC значения $\max |\delta|$

выше, причем на малых скоростях наиболее активное управление формирует MPC (13.767° при 3 м/с и 8.431° при 5 м/с). Аналогичная закономерность наблюдается и по метрике $STD(\delta)$: на всех скоростях наименьший разброс управляющего воздействия соответствует PP, тогда как LQR и MPC формируют более вариативное управление. При $\bar{v} = 10$ м/с MPC обеспечивает заметно меньшие значения $\max|\delta|$ и $STD(\delta)$, чем LQR; при одинаковом значении $RMSE(e_y)$ для этих регуляторов MPC также характеризуется меньшим 95%-квантилем ошибки.

По вычислительным затратам все три алгоритма удовлетворяют требованиям реального времени при шаге дискретизации $T_s = 0.1$ с. Регулятор PP требует в среднем 0.021 мс на шаг, регулятор LQR — от 0.026 до 0.052 мс на шаг. Для MPC среднее время вычисления составляет от 2.386 до 2.703 мс, а 95%-квантиль находится в диапазоне 4.590–5.053 мс. Следовательно, решение задачи квадратичного программирования приводит к увеличению вычислительной нагрузки примерно на два порядка по сравнению с PP и LQR, однако полученный запас по времени остается достаточным для практической реализации в реальном времени на выбранной платформе.

Для сопоставления алгоритмов по критериям, значимым для дальнейшего выбора базового регулятора, целесообразно использовать сводную качественную оценку, приведенную в таблицу 8. Столбцы, характеризующие учет ограничений и ограничения по поперечному ускорению, заполнены по структурным свойствам алгоритмов, поскольку в данной серии вычислительных экспериментов прямые численные метрики по поперечному ускорению отдельно не выделялись.

Таблица 8 – Сопоставление регуляторов по основным критериям

Алгоритм	Точность слежения	Учет ограничений	Ограничение a_y
MPC	Приемлемая на всех скоростях; при 10 м/с одинаковая с LQR RMSE и меньший 95%-квантиль	явный	явный
LQR	Наилучшая при 3, 5 и 13.88 м/с; при 10 м/с уступает PP по RMSE и MPC по 95%-квантилю	косвенный	нет
PP	Высокая при 10 и 13.88 м/с; лучший результат при 10 м/с	нет	нет

Из представленных результатов следует, что в рассматриваемом вычислительном эксперименте регулятор LQR обеспечивает наилучшую точность поперечного слежения в большинстве исследованных режимов. Регулятор PP формирует наиболее мягкое управляющее воздействие и требует минимальных вычислительных затрат. Регулятор MPC не обеспечивает

выигрыша по точности на фиксированной S-образной траектории, но сохраняет приемлемое качество слежения и вычислительную реализуемость в реальном времени.

Выбор базового алгоритма для дальнейшей работы определяется не только минимизацией поперечной ошибки на частной синтетической трассе, но и соответствием регулятора постановке задачи движения в условиях ограничений. С этой точки зрения модельно-предиктивное управление обладает принципиальным преимуществом, поскольку допускает явное включение ограничений на угол поворота рулевого колеса, скорость его изменения, поперечное ускорение и другие динамические ограничения в единую оптимизационную постановку. Для PP и LQR такой учет в базовой схеме отсутствует и реализуется только косвенно.

LQR следует рассматривать как регулятор, обеспечивающий наилучшую точность слежения в данном частном вычислительном эксперименте. MPC следует рассматривать как регулятор, в наибольшей степени соответствующий архитектуре последующей системы управления движением, ориентированной на явный учет ограничений. По этой причине в дальнейшей работе в качестве базового алгоритма следования траектории принимается модельно-предиктивное управление.

2.6 Выводы

В главе 2 сформирована модельная и алгоритмическая основа для последующей разработки и экспериментальной апробации методов планирования и управления движением беспилотного автомобиля. Рассмотрено представление модели автомобиля в пространстве состояний и выполнена линеаризация нелинейных уравнений движения, что позволило получить линейные модели ошибок слежения, пригодные для синтеза оптимальных регуляторов. В целях обеспечения гладкости и дифференцируемости опорной траектории выполнена ее аппроксимация кубическим B-сплайном, обеспечивающая вычислительно устойчивое получение геометрических характеристик траектории (включая курс и кривизну) при дальнейших расчетах.

В рамках единой методики сопоставления исследованы три подхода к следованию траектории: геометрический регулятор Pure Pursuit, линейно-квадратичный регулятор и модельно-предиктивное управление. Для

исключения методической неоднозначности в явном виде разделены модель объекта управления, используемая в симуляции для обновления состояния, и внутренняя модель регулятора, применяемая при синтезе управления; показано, что такая постановка корректно описывает типичную ситуацию модельного рассогласования и позволяет интерпретировать результаты как сравнение регуляторов при фиксированном объекте управления и фиксированной процедуре вычисления ошибок.

Результаты моделирования на S-образной траектории при опорных скоростях $\bar{v} \in \{3, 5, 10, 13.88\}$ м/с показали, что при выбранных настройках LQR обеспечивает наименьшую $\text{RMSE}(e_y)$ в трех из четырех режимов, тогда как при 10 м/с наилучший результат получен для Pure Pursuit. Победитель по метрике $q_{0.95}(|e_y|)$ также зависит от скорости. Pure Pursuit обеспечивает приемлемое качество при минимальной вычислительной сложности и может рассматриваться как базовый алгоритм для задач, не требующих явного учета ограничений. Модельно-предиктивное управление, приведенное к задаче квадратичного программирования, сохраняет вычислительную реализуемость в реальном времени при принятом шаге дискретизации, однако требует большего вычислительного времени на такт вследствие решения задачи оптимизации. Несмотря на отсутствие безусловного выигрыша MPC по точности на фиксированной тестовой трассе, в дальнейшей работе в качестве базового алгоритма следования траектории выбран модельно-предиктивный подход. Данное решение обусловлено тем, что критерием выбора регулятора в задаче управления беспилотным автомобилем является не только среднеквадратическая точность слежения, но и возможность обеспечения ограничений и требований безопасности в широком диапазоне дорожных сценариев, включая режимы, близкие к насыщению исполнительных механизмов и границам устойчивости движения.

ГЛАВА 3. Разработка алгоритма планирования траектории с учетом дорожной обстановки

В данной главе рассматривается последовательная разработка алгоритма локального планирования траектории движения автомобиля с учетом дорожной обстановки. На первом этапе исследуется задача сглаживания опорной траектории без учета ограничений, накладываемых свободным пространством, что позволяет выделить влияние целевой функции и параметров оптимизации на плавность траектории. На втором этапе в постановку вводятся ограничения, обусловленные границами проезжей части, благодаря чему обеспечивается построение траектории в допустимом коридоре движения. На третьем этапе формулируется задача выпуклой оптимизации в системе координат Френе с учетом габаритов автомобиля, геометрии дороги и статических препятствий.

Решаемая в настоящей главе задача состоит в построении траектории, которая одновременно удовлетворяет трем основным требованиям: остается в пределах допустимой области движения, является гладкой и динамически реализуемой для автомобиля, а также может быть вычислена за ограниченное время, допустимое для работы бортовой системы управления. Результаты каждого этапа оцениваются не только с точки зрения геометрической корректности траектории, но и с точки зрения понятных прикладных признаков: уменьшения резких изменений направления движения, соблюдения дорожных ограничений и возможности применения алгоритма в режиме, близком к реальному времени. В настоящей главе рассматривается формирование локальной траектории движения беспилотного транспортного средства на основе методов выпуклой оптимизации с учетом дорожных ограничений и элементов дорожной обстановки. Необходимость указанного этапа обусловлена тем, что исходная опорная линия (центральная линия полосы/маршрута), получаемая из цифровой карты либо результатов восприятия, как правило, задана дискретно и характеризуется разрывами или высокочастотными колебаниями кривизны. Это приводит к ухудшению исполнимости траектории в контуре управления и увеличению управляющих воздействий, а также к росту динамических показателей, связанных с комфортом движения.

В качестве математической основы используются модели движения и

преобразования к задачам квадратичного программирования, введенные в главе 2, включая представление траектории и ошибок слежения в криволинейной системе координат Френе, а также QR-постановки, применимые для решения задач сглаживания и планирования в режиме реального времени. Это обеспечивает единообразие используемых обозначений и согласованность принятых допущений при переходе от этапа моделирования и синтеза регуляторов к этапу генерации траектории.

Блок-схема алгоритма генерации локальной траектории движения автомобиля с учетом дорожных ограничений по маршруту движения представлена на рисунке 18.



Рисунок 18 – Блок-схема алгоритма генерации локальной траектории движения автомобиля с учетом дорожных ограничений по маршруту движения представлена

Далее в разделе 3.1 формулируется базовая задача сглаживания опорной траектории в декартовой системе координат без учета свободного пространства. В разделе 3.2 рассматривается расширение указанной постановки за счет введения ограничений на границы дороги. В разделе 3.3 приводится метод эффективного планирования траектории в системе координат Френе с явным учетом границ дороги и геометрии транспортного средства, а также описываются приемы ускорения решения QR и модификации, обеспечивающие учет статических препятствий.

При извлечении центральной линии, используемой в качестве исходного

приближения при генерации траектории движения беспилотного транспортного средства, опорная кривая, как правило, задается дискретным набором точек и характеризуется недостаточной гладкостью. Это проявляется в неплавном изменении кривизны κ и ее производной по дуговой координате, что потенциально приводит к резким изменениям требуемого рулевого воздействия при последующем исполнении траектории.

Опорная (референсная) траектория далее представляется последовательностью точек с декартовыми координатами (x, y) . Для вычисления кривизны траектория параметризуется дуговой координатой s (длиной дуги кривой), которая является естественным параметром движения вдоль линии пути и не зависит от времени. В общем случае для гладкой кривой $r(t)$ интегральное выражение для длины дуги на интервале $t \in [t_0, t_1]$ имеет вид

$$r(t) = (x(t), y(t), z(t)), \quad t \in [t_0, t_1], \quad (3.1)$$

$$s(t_1) = \int_{t_0}^{t_1} |r'(t)| dt = \int_{t_0}^{t_1} \sqrt{\sum_{k=1}^n \dot{r}_k(t)^2} dt. \quad (3.2)$$

Поскольку в рассматриваемой задаче траектория задана дискретным набором точек, дуговая координата оценивается численно. Для соседних точек вычисляются приращения

$$\Delta x_i = x_{i+1} - x_i, \quad (3.3)$$

$$\Delta y_i = y_{i+1} - y_i, \quad (3.4)$$

$$\Delta s_i = \sqrt{(\Delta x_i)^2 + (\Delta y_i)^2}, \quad (3.5)$$

после чего выполняется накопление длины дуги вдоль траектории:

$$s_0 = 0, \quad s_{i+1} = s_i + \Delta s_i. \quad (3.6)$$

Тем самым формула (3.5) задает процедуру вычисления дуговой координаты для каждой дискретной точки.

Кривизна траектории, параметризованной дуговой координатой s , вычисляется по соотношению

$$\kappa(s) = \frac{x'(s) y''(s) - y'(s) x''(s)}{(x'(s)^2 + y'(s)^2)^{3/2}}. \quad (3.7)$$

Производные $x'(s)$ и $x''(s)$ (аналогично для $y'(s)$ и $y''(s)$) оцениваются по дискретным данным с использованием разностных аппроксимаций:

$$x'(s_i) \approx \frac{x_{i+1} - x_{i-1}}{s_{i+1} - s_{i-1}}, \quad (3.8)$$

$$x''(s_i) \approx \frac{\frac{x_{i+1} - x_i}{s_{i+1} - s_i} - \frac{x_i - x_{i-1}}{s_i - s_{i-1}}}{\frac{1}{2}[(s_{i+1} - s_i) + (s_i - s_{i-1})]}. \quad (3.9)$$

Построенные по указанной процедуре зависимости показывают, что траектория, извлеченная из карты, сопровождается скачкообразным характером изменения кривизны (см. рисунок 19).

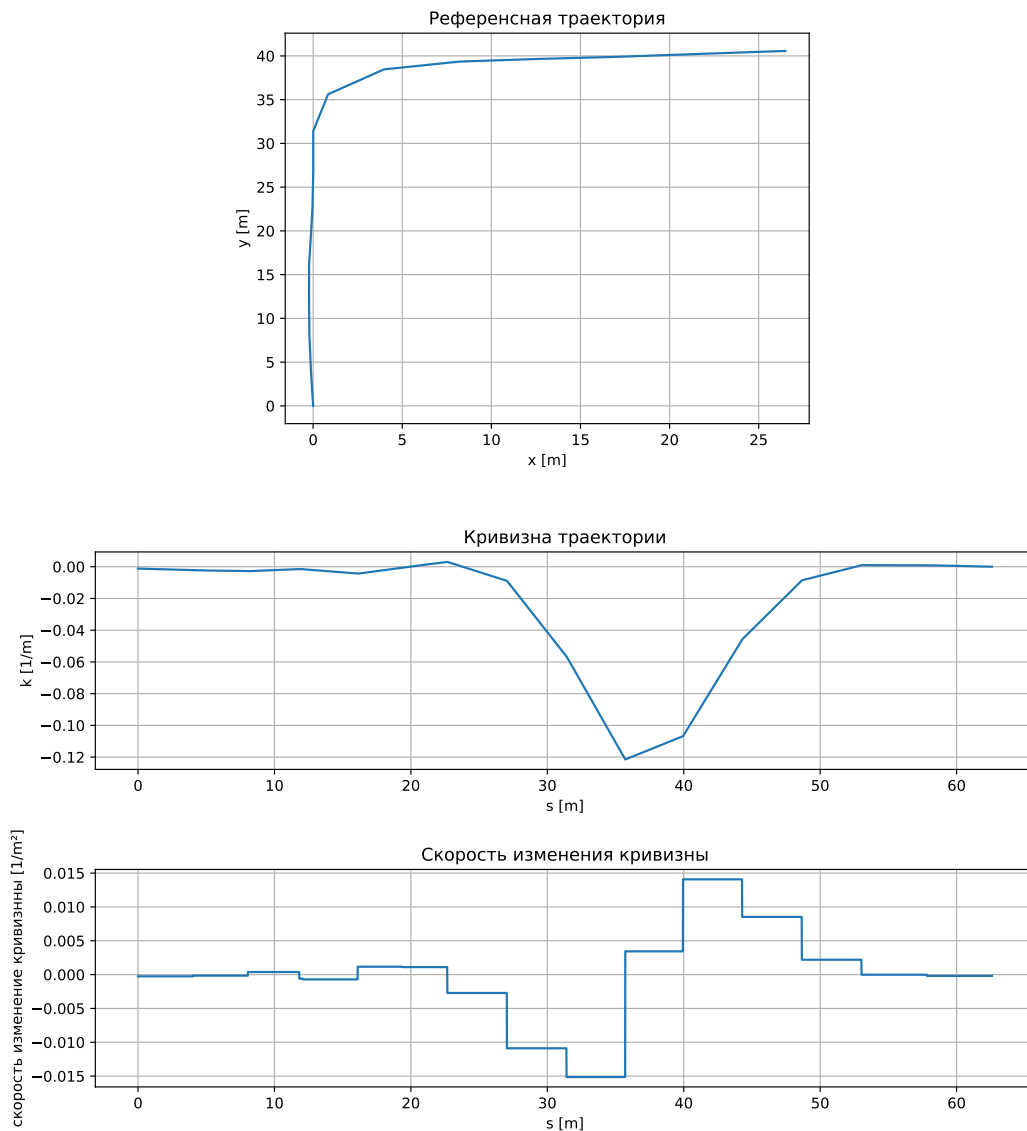


Рисунок 19 – Извлеченная референсная траектория и графики ее кривизны и скорости изменения кривизны построенные относительно пройденного пути

Скачкообразный характер изменения $\kappa(s)$ и $\kappa'(s)$, наблюдаемый для траектории, извлеченной из карты (рисунок 19), указывает на недостаточную гладкость исходных данных и, как следствие, на потенциальное ухудшение исполнимости траектории в контуре управления. В частности, разрывы и высокочастотные колебания кривизны приводят к резким изменениям требуемого рулевого воздействия и могут снижать устойчивость слежения.

Для формирования опорной траектории, пригодной для последующего оптимизационного сглаживания и исполнения, в настоящей работе применяется двухэтапная процедура предварительной обработки. На первом этапе выполняется аппроксимация дискретной опорной линии кубическим B -сплайном, что обеспечивает повышенную гладкость и непрерывность кривизны (вплоть до C^2 -непрерывности) и позволяет вычислять κ и, при необходимости, ее производную. На втором этапе полученная траектория дополнительно корректируется путем решения задачи квадратичного программирования, в целевой функции которой явно минимизируются отклонение от опорной линии, величину кривизны и скорость ее изменения.

3.1 Алгоритм сглаживания траектории движения беспилотного транспортного средства на основе квадратичной оптимизации без учета свободного пространства

3.1.1 Постановка задачи сглаживания траектории без учета свободного пространства

В настоящем разделе рассматривается задача сглаживания опорной (референсной) траектории движения беспилотного транспортного средства (БТС) без учета свободного пространства, то есть без явного учета границ дорожной полосы и препятствий. Целью сглаживания является формирование геометрически более плавной траектории, характеризующейся меньшими значениями кривизны и скорости изменения кривизны при сохранении близости к исходной опорной линии.

Входными данными являются дискретная опорная траектория, заданная последовательностью точек $\{x_{ref,i}, y_{ref,i}\}_{i=0}^{N-1}$, а также связанные с ней опорные значения ориентации $\theta_{ref,i}$ и кривизны $k_{ref,i}$ (при наличии), вычисляемые по процедуре параметризации траектории дуговой координатой и разностной

аппроксимации производных, приведенной во вводной части главы. Результатом является сглаженная дискретная траектория $\{x_i, y_i, \theta_i, k_i\}_{i=0}^{N-1}$, предназначенная для дальнейшего использования в последующих постановках, учитывающих дорожные ограничения [68].

3.1.2 Формат представления траектории в декартовой системе координат и дискретная модель

Траектория в декартовой системе координат представляется последовательностью состояний, включающих положение, ориентацию и кривизну:

$$\mathbf{x}(s) = \begin{bmatrix} x(s) & y(s) & \theta(s) & k(s) \end{bmatrix}^T,$$

где s — продольная (дуговая) координата вдоль траектории.

Дискретная траектория задается на сетке $\{s_i\}_{i=0}^{N-1}$ с шагом $ds = s_{i+1} - s_i$, что соответствует равномерной дискретизации вдоль пути. Для наглядности дискретное представление траектории приведено на рисунке 20.

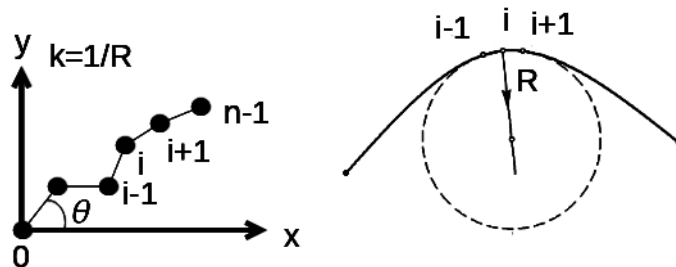


Рисунок 20 – Дискретное представление траектории

В качестве кинематической модели в пространстве состояний используется зависимость производных по дуговой координате:

$$\begin{bmatrix} x' \\ y' \\ \theta' \\ k' \end{bmatrix} = \begin{bmatrix} \cos(\theta) \\ \sin(\theta) \\ k \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u, \quad (3.10)$$

где $u = k'$ — управляющее воздействие, интерпретируемое как скорость изменения кривизны вдоль траектории; штрих $(\cdot)'$ обозначает производную по s .

3.1.3 Предварительная аппроксимация опорной траектории В-сплайном

Поскольку исходная опорная траектория, как правило, задается дискретно и может обладать недостаточной гладкостью, перед решением задачи квадратичного программирования выполняется предварительная аппроксимация опорной линии кубическим В-сплайном. Указанный этап обеспечивает повышение гладкости геометрических характеристик траектории, в том числе непрерывность кривизны и ее производной, что снижает вероятность появления резких изменений требуемого рулевого воздействия при последующем исполнении траектории (рисунок 21).

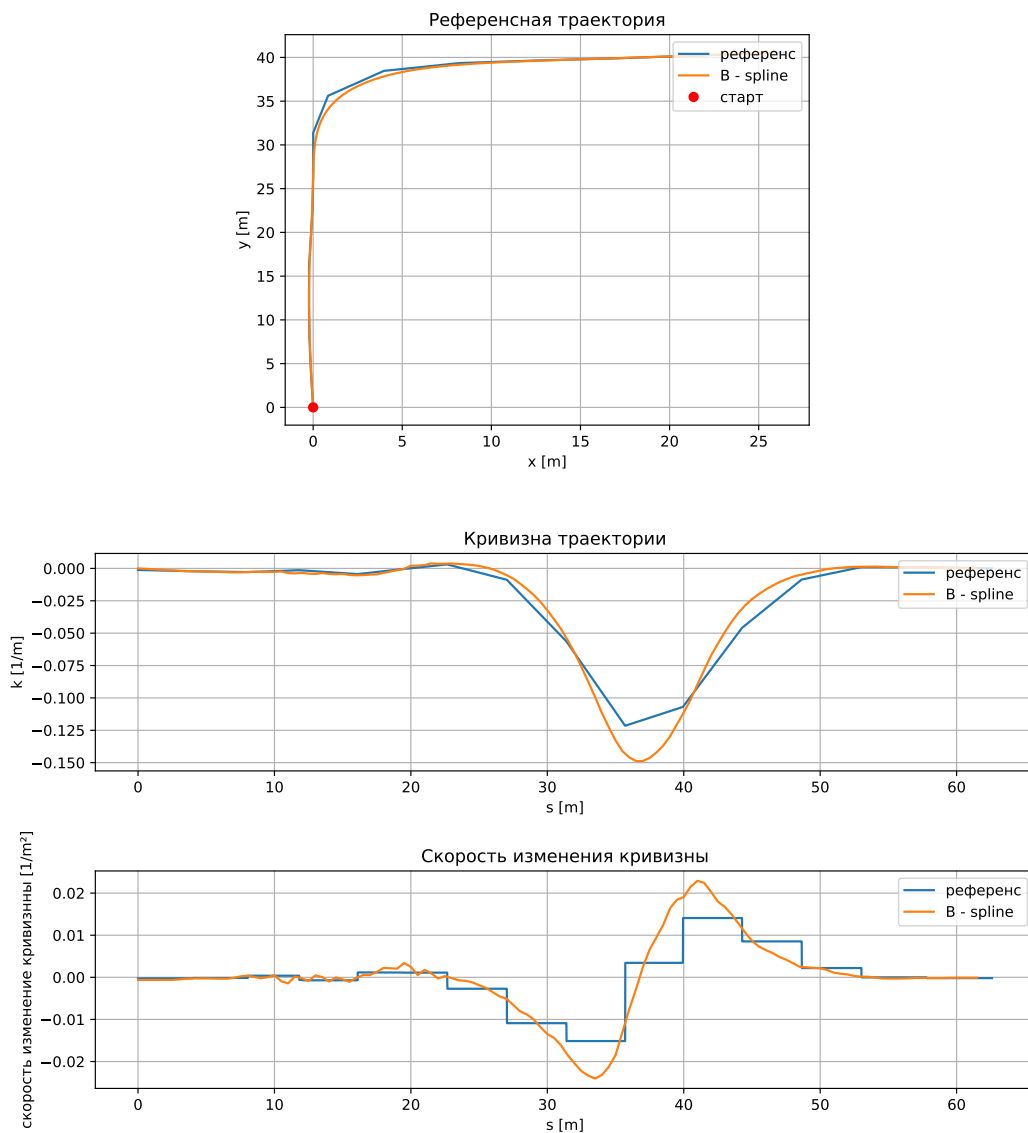


Рисунок 21 – Сравнение изначальной и аппроксимированной с помощью В-сплайна траектории

Как видно из рисунка 21, после аппроксимации B -сплайном зависимости k и k' приобретают непрерывный характер. Полученная траектория рассматривается далее как опорная для последующего оптимизационного сглаживания.

3.1.4 Целевая функция задачи квадратичного программирования

Задача сглаживания формализуется как задача квадратичного программирования, в которой минимизируется отклонение от опорной траектории, а также штрафуются величина кривизны и ее изменения вдоль пути. В рамках принятого представления 3.1.2 целевая функция задается выражением:

$$J = \sum_{i=0}^{N-1} \left(W_d (x_i - x_{ref,i})^2 + W_d (y_i - y_{ref,i})^2 + W_c k_i^2 \right) + \sum_{i=0}^{N-2} W_{cr} (k_{i+1} - k_i)^2, \quad (3.11)$$

где W_d — весовой коэффициент, определяющий степень следования опорной траектории; W_c — весовой коэффициент, штрафующий величину кривизны; W_{cr} — весовой коэффициент, штрафующий изменения кривизны между соседними узлами дискретизации (дискретный аналог производной кривизны).

3.1.5 Ограничения и линеаризация динамики

Для приведения задачи к квадратичному программированию нелинейные соотношения динамики (3.10) линеаризуются в окрестности опорных точек. Линеаризация выполняется по формуле Тейлора первого порядка формула 2.16. В результате для дискретной модели с шагом ds получаются линейные соотношения вида:

$$\begin{aligned} x_{i+1} &= x_i + ds \cdot \cos(\theta_{ref,i}) - \theta_i \cdot ds \cdot \sin(\theta_{ref,i}), \\ y_{i+1} &= y_i + ds \cdot \sin(\theta_{ref,i}) + \theta_i \cdot ds \cdot \cos(\theta_{ref,i}), \\ \theta_{i+1} &= \theta_i - ds \cdot k_{ref} + ds \cdot k_i. \end{aligned} \quad (3.12)$$

Планирование начинается из первой (индекс 0) опорной точки траектории. Для обеспечения согласованности начала сглаженной траектории с заданной

стартовой позицией вводятся начальные условия:

$$\begin{aligned} x_0 &= x_{ref,0}, \\ y_0 &= y_{ref,0}. \end{aligned} \quad (3.13)$$

3.1.6 Результаты моделирования и обсуждение влияния весов

На рисунке 22 приведен пример результата сглаживания, выполненного в два этапа: предварительная аппроксимация B -сплайном и последующее решение задачи квадратичного программирования.

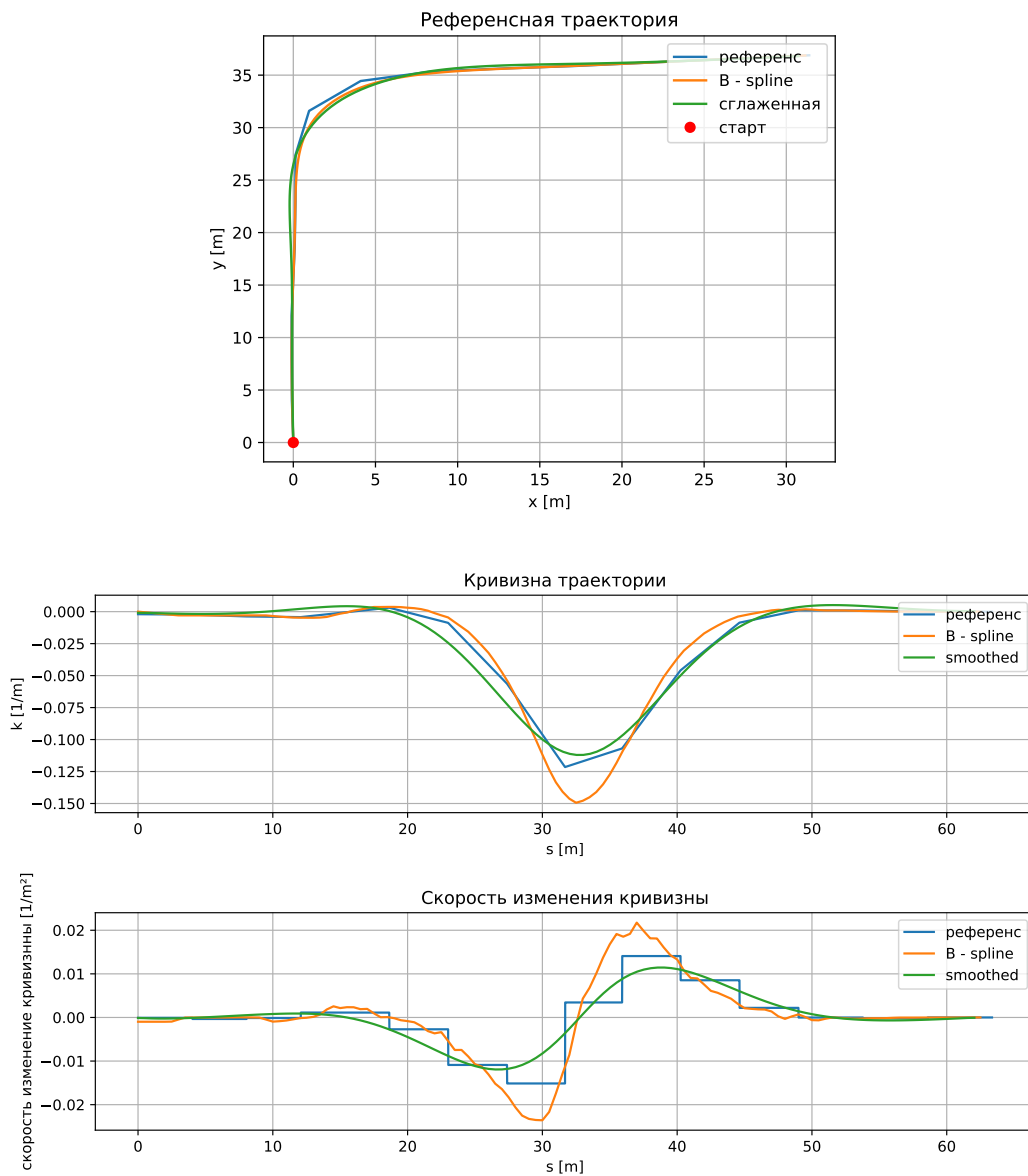


Рисунок 22 – Результат работы алгоритма сглаживания при $W_d = 0.005$; $W_c = 1.0$; $W_{cr} = 10.0$

Как видно из рисунка 22, оптимизационное сглаживание обеспечивает повышение плавности траектории по сравнению с вариантом, полученным только после аппроксимации B -сплайном, что выражается в снижении амплитуды колебаний кривизны и ее изменения вдоль пути.

Таблица 9 – Сравнение траекторий по данным рис. 3.19 ($W_d = 0.005$, $W_c = 1.0$, $W_{cr} = 10.0$)

Показатель	Референс	B-spline	Сглаженная
Длина по дуге, $s_{\max}$, м	65	65	65
Минимальная кривизна, $k_{\min}$, 1/м	-0.125	-0.155	-0.110
Положение минимума, $s(k_{\min})$, м	32	33	34
Максимальная кривизна, $k_{\max}$, 1/м	+0.002	+0.003	+0.002
Максимальная скорость изменения кривизны, $\max dk/ds $, 1/м ²	0.015	0.023	0.012
Характер dk/ds (качественно)	ступенчатый	колебательный	сглаженный

При нулевых значениях весов, отвечающих за минимизацию кривизны и изменений кривизны ($W_c = 0$, $W_{cr} = 0$), решение задачи вырождается в следование опорной траектории, то есть оптимизация не вносит сглаживающих коррекций в геометрию траектории.

Таким образом, предложенная постановка позволяет управлять компромиссом между точностью следования опорной траектории и требованиями к плавности за счет изменения весовых коэффициентов W_d , W_c и W_{cr} . Вместе с тем рассматриваемая постановка не содержит ограничений, связанных с границами дорожной полосы и наличием препятствий. Вследствие этого сглаженная траектория может выходить за пределы полосы движения либо пересекать препятствия, что ограничивает применимость алгоритма в задачах безопасного планирования.

Для устранения указанного недостатка далее рассматривается постановка, в которой сглаживание выполняется с учетом ограничений на границы дороги и доступное пространство движения.

3.2 Алгоритм сглаживания траектории движения беспилотного транспортного средства на основе квадратичной оптимизации с учетом ограничений на границы дороги

В разделе 3.1 была рассмотрена задача сглаживания опорной траектории без учета свободного пространства. На практике такой подход недостаточен, поскольку результирующая траектория может выходить за пределы полосы движения либо пересекать области, занятые препятствиями, что делает ее непригодной для последующего исполнения в контуре управления.

Настоящий раздел направлен на расширение постановки за счет явного учета геометрических ограничений в декартовой системе координат [69]. При этом рассматривается упрощенная модель, в которой транспортное средство аппроксимируется материальной точкой, а ограничения на допустимую область движения задаются в виде коридора.

3.2.1 Формулировка задачи сглаживания траектории с учетом ограничений

Пусть задана опорная траектория в виде дискретного набора точек $\{(x_{ref,i}, y_{ref,i})\}_{i=0}^{N-1}$, полученная, например, методами дискретного планирования (поиск на графах и др.). Требуется построить сглаженную траекторию, обеспечивающую малое отклонение от опорной линии, ограничение кривизны и ограничение скорости изменения кривизны, а также удовлетворяющую ограничениям на допустимую область движения.

В отличие от раздела 3.1.2, дополнительно вводятся ограничения, задающие коридор движения в каждой дискретной точке траектории. В рамках данной постановки предполагается, что коридор может быть задан как интервальные ограничения по координате y при фиксированной (или монотонно возрастающей) координате x . Данное допущение вводится для получения линейных ограничений и, тем самым, возможности решения задачи методами квадратичного программирования в режиме реального времени.

3.2.2 Представление траектории и формирование ограничений на свободное пространство

Модель траектории в декартовой системе координат приведена в разделе 3.1.2. Однако в общем случае границы проезжей части в декартовой системе координат имеют нелинейный характер (например, при развороте дороги на 180°

или при самопересечении опорной траектории, рисунок 23).

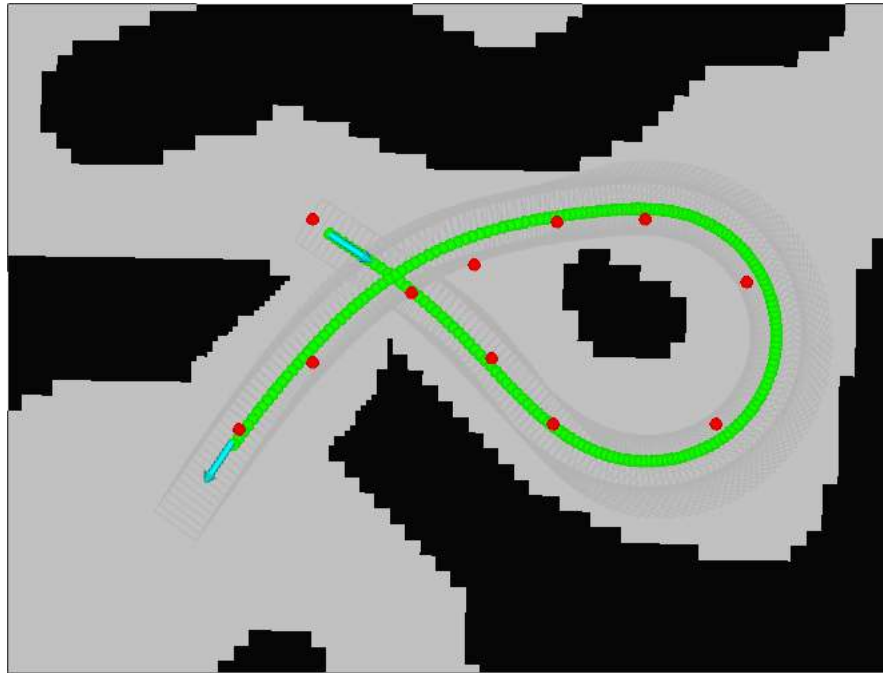


Рисунок 23 – Пример самопересечения опорной траектории

Для перехода к линейным ограничениям используется параметризация по координате x (рисунок 24), при которой x рассматривается как заданная (индексируемая) величина, а оптимизируемыми переменными являются y , θ и k .

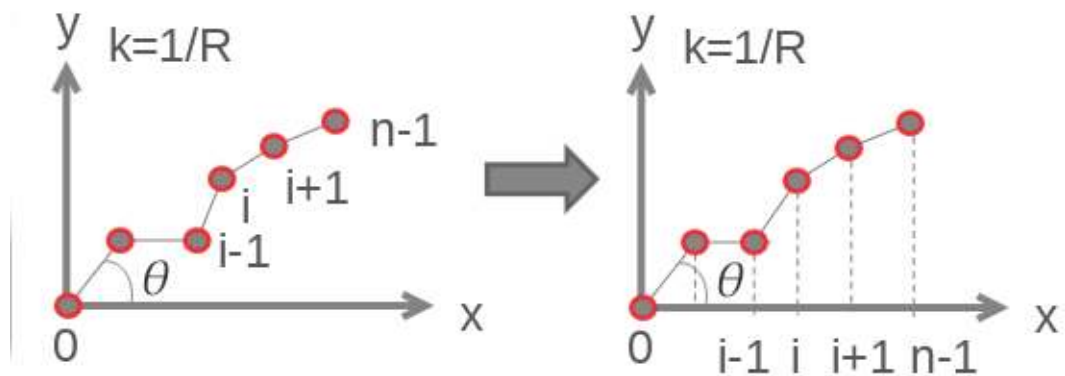


Рисунок 24 – Дискретное представление траектории (слева: x как переменная, справа: x задан индексом)

$$\begin{bmatrix} y' \\ \theta' \\ k' \end{bmatrix} = \begin{bmatrix} \sin(\theta) \\ k \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u, \quad (3.14)$$

где u — управляющая величина, соответствующая скорости изменения кривизны ($u \equiv k'$).

Исключение координаты x из уравнений динамики эквивалентно введению допущения о монотонном продвижении вдоль оси x (то есть без участков движения, требующих уменьшения x). В рамках данного допущения границы допустимой области движения могут быть заданы линейно в виде интервальных ограничений на y в каждой дискретной точке:

$$lb_i \leq y_i \leq ub_i, \quad i = 0, \dots, N - 1, \quad (3.15)$$

где lb_i и ub_i — соответственно нижняя и верхняя границы коридора движения, заданные по данным карты и (при наличии) статических препятствий.

На рисунке 25 приведена иллюстрация формирования коридора проходимости для случая, когда исходная опорная линия пересекает область, частично занятые статическими препятствиями, и требует корректировки для сохранения допустимых зазоров.

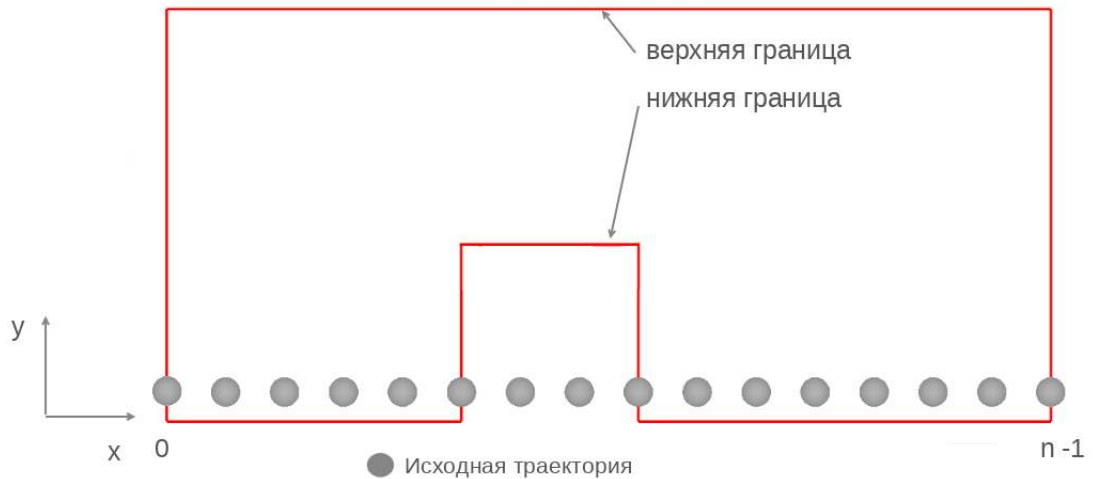


Рисунок 25 – Визуализация ограничений вида формула (3.15)

На рисунке 25 показан характер локальной корректировки коридора проходимости при наличии статического препятствия на исходной опорной линии. В области конфликта с препятствием границы коридора сужаются и/или смещаются таким образом, чтобы сохранялся заданный безопасный зазор, а допустимая область оставалась связной. Для каждого дискретного узла траектории $i = 0, \dots, n-1$ формируются верхняя и нижняя границы допустимого поперечного отклонения, которые далее используются в виде неравенств 3.15. Тем самым обеспечивается исключение пересечений с занятыми областями при сохранении исходного коридора на участках, свободных от препятствий, и создаются корректные ограничения для последующей оптимизации траектории.

3.2.3 Целевая функция задачи квадратичного программирования

Для задачи сглаживания с учетом ограничений (3.15) используется квадратичный функционал качества:

$$J = \sum_{i=0}^{N-1} \left(W_d (y_i - y_{ref,i})^2 + W_c k_i^2 \right) + \sum_{i=0}^{N-2} W_{cr} (k_{i+1} - k_i)^2, \quad (3.16)$$

где W_d — вес отклонения от опорной траектории, W_c — вес регуляризации кривизны, W_{cr} — вес регуляризации скорости изменения кривизны.

3.2.4 Ограничения и линеаризация динамики

С помощью ряда Тейлора 2.16 уравнения (3.14) линеаризуются в окрестности опорных значений:

$$\begin{aligned} y_{i+1} &= y_i + ds \cdot \sin(\theta_{ref,i}) + \theta_i \cdot ds \cdot \cos(\theta_{ref,i}), \\ \theta_{i+1} &= \theta_i - ds \cdot k_{ref} + ds \cdot k_i, \end{aligned} \quad (3.17)$$

где ds — шаг дискретизации, согласованный с дискретизацией опорной траектории.

Начальная точка фиксируется в качестве начального условия:

$$\begin{aligned} x_0 &= x_{ref,0}, \\ y_0 &= y_{ref,0}. \end{aligned} \quad (3.18)$$

3.2.5 Численный эксперимент и обсуждение

В рассматриваемом примере ограничения, формула (3.15) вынуждают оптимизатор отклоняться от опорной линии в тех участках, где ее следование приводит к нарушению границ коридора или к пересечению областей, занятых препятствиями (рисунок 26). Тем самым обеспечивается совместное выполнение требований к плавности и геометрической допустимости траектории при заданной структуре ограничений.

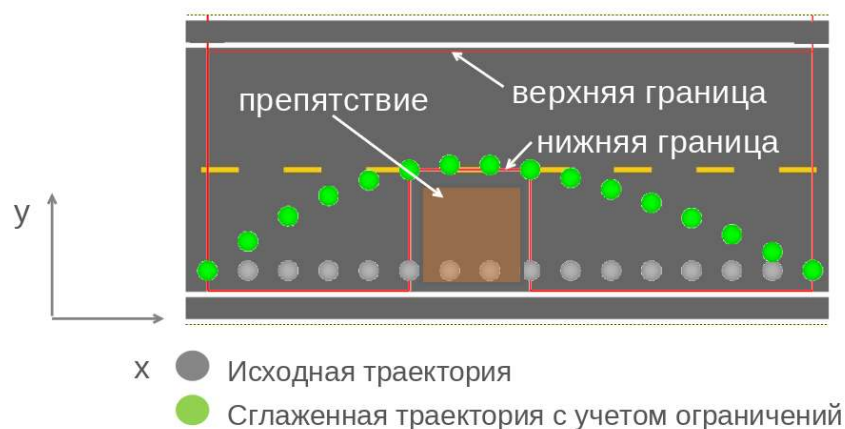


Рисунок 26 – Пример корректировки траектории при заданном коридоре движения (параметризация по координате x)

Основная сводка по решению задачи оптимизации представлена в таблице 10.

Таблица 10 – Основная сводка по решению заданной задачи оптимизации

Число переменных	50
Число ограничений	49
Используемый решатель СЛАУ	qdlldl
Время выполнения	0.16 мс

Полученная оценка времени решения подтверждает возможность применения постановки в задачах, предъявляющих требования к работе в режиме реального времени.

Вместе с тем предложенная параметризация по координате x ограничивает область применимости: коридор должен быть однозначно представим в виде интервальных ограничений по y для каждого x_i , исключаются случаи разворотов, самопересечений и иных конфигураций, при которых движение не является монотонным по x , геометрия транспортного средства (габариты) в данной постановке не учитывается. Указанные ограничения являются мотивирующими предпосылками для перехода к постановке в системе координат Френе, где ограничения по границам дороги формулируются естественным образом относительно опорной линии и допускают учет габаритов транспортного средства.

3.3 Эффективное планирование траектории для автономных транспортных средств с использованием выпуклой оптимизации в системе координат Френе с учетом границ дороги

Параметризация траектории по одной из декартовых координат, рассмотренная в разделе 3.2.2, корректна только при монотонном движении по выбранной оси и, как следствие, ограничивает область применимости постановки при маневрах на участках со сложной геометрией (повороты, развороты, участки с обратным ходом по координате). Для устранения указанного ограничения далее используется криволинейная система координат Френе, связанная с опорной линией движения.

3.3.1 Представление в системе координат Френе

Переход к координатам Френе и принятые далее обозначения (дуговая координата s , поперечное отклонение d , угловая ошибка e_θ , опорные величины $k_{ref}(s)$ и $k'_{ref}(s)$) введены в главе 2 при выводе поперечной кинематической модели в криволинейных координатах. В частности, в главе 2 определена связь временной и пространственной параметризации через величину $\dot{s}$ (см. (2.52)), записана нелинейная модель $z'(s)$ (см. (2.54)) и получена ее линеаризация в окрестности опорной траектории (см. (2.57)). В настоящем разделе указанные соотношения используются как исходные при построении выпуклой постановки квадратичного программирования, позволяющей учитывать границы проезжей части и габариты транспортного средства в виде линейных ограничений.

Вектор состояния в системе координат Френе записывается в формате:

$$[d, e_\theta, k]^\top \quad (3.19)$$

Где d — поперечное отклонение относительно опорной линии (вдоль нормали к опорной траектории), e_θ — угловая ошибка (разность курсов между направлением движения и касательной к опорной траектории), k — кривизна планируемой траектории.

В качестве переменной управления будет использоваться k' - скорость изменения кривизны.

Далее, как и в главе 2, штрих $(\cdot)'$ обозначает производную по дуговой координате s ; при необходимости переход к временной производной осуществляется по формуле $\dot{z} = z' \dot{s}$ с использованием соотношения (2.52).

Поперечная кинематическая модель транспортного средства в координатах Френе получена в главе 2 (см. (2.54)) и для полноты изложения приведена ниже [70]:

$$z' = \begin{bmatrix} d' \\ e'_\theta \\ k' \end{bmatrix} = \begin{bmatrix} (1 - k_{ref}d) \tan(e_\theta) \\ \frac{(1 - k_{ref}d)k}{\cos(e_\theta)} - k_{ref} \\ u \end{bmatrix} \quad (3.20)$$

Линеаризация выполняется в окрестности опорной траектории, то есть при

$d = 0, e_\theta = 0, k = k_{ref}$ и $k' = k'_{ref}$.

$$z' = Az + Bu \quad (3.21)$$

$$z' = \begin{bmatrix} 0 \\ 0 \\ k'_{ref} \end{bmatrix} + \begin{bmatrix} 0 & 1 & 0 \\ -k_{ref}^2 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} d \\ e_\theta \\ k - k_{ref} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} (k' - k'_{ref}) \quad (3.22)$$

Полученная линейная модель эквивалентна выражению (2.57), приведенному в главе 2.

Заменяя производную по дуговой координате разностной аппроксимацией по шагу $ds = s_{i+1} - s_i$ и вводя вектор отклонений

$$\Delta z_i = \begin{bmatrix} d_i & e_{\theta,i} & k_i - k_{ref,i} \end{bmatrix}^T, \quad \Delta u_i = k'_i - k'_{ref,i}, \quad (3.23)$$

получаем дискретную модель в виде

$$\Delta z_{i+1} = (I + ds A) \Delta z_i + ds B \Delta u_i, \quad (3.24)$$

где матрицы A и B определяются выражением (2.57) (глава 2). В дальнейшем ограничениями и функционалом качества задаются требования к Δz_i и Δu_i , а абсолютные значения кривизны восстанавливаются как $k_i = (k_i - k_{ref,i}) + k_{ref,i}$.

Далее, вводятся ограничения на кривизну и на скорость изменения кривизны, чтобы гарантировать проходимость траектории на заданной скорости.

На низких скоростях ($v < 5$ м/с) ограничение кривизны задается через максимальный угол поворота руля, извлеченный из велосипедной модели.

$$k \in [-\tan(\delta_{max})/L, +\tan(\delta_{max})/L] \quad (3.25)$$

Где δ_{max} — максимально допустимый угол поворота управляемых колес, L — колесная база.

Для больших скоростей ($v \geq 5$ м/с) вводится зависимость кривизны от скорости автомобиля, а также максимально допустимого бокового ускорения в заданных условиях.

$$k_{max} = \frac{a_{y_{max}}}{v^2} \quad (3.26)$$

Где $a_{y_{max}}$ - максимально допустимое боковое ускорение, v - скорость автомобиля.

Максимально допустимое боковое ускорение с учетом сцепления шин с дорожным покрытием принимается по соотношению:

$$a_{y_{max}} = \mu g \quad (3.27)$$

Где μ - коэффициент сцепления с дорогой, g - ускорение свободного падения.

Для ограничения бокового ускорения с учетом сцепления шин с дорожным покрытием принимается коэффициент трения μ в диапазоне от 0.7 до 0.9 для сухого асфальта и от 0.4 до 0.6 для влажного городского покрытия [66, 71, 72]. Указанные значения соответствуют отраслевым стандартам и подтверждаются экспериментальными исследованиями.

В следующей секции представлен метод формирования ограничений на геометрию автомобиля для квадратичной проблемы.

3.3.2 Формирование полигона проходимости и учет геометрии автомобиля

Для учета границ проезжей части и статических препятствий разработан многоэтапный метод формирования полигона проходимости в системе координат Френе. Метод включает аппроксимацию исходной опорной линии B -сплайном; формирование латерально дискретизированного графа; поиск алгоритмом A^* связного бесколлизийного пути; сглаживание найденного пути; аппроксимацию габаритов автомобиля набором покрывающих окружностей; определение по карте дистанций верхних и нижних границ допустимого коридора; включение полученных границ в систему линейных ограничений задачи квадратичного программирования.

Графовый поиск определяет допустимый вариант прохождения относительно статических препятствий, тогда как последующая квадратичная оптимизация обеспечивает геометрическую гладкость и выполнение кинематических ограничений. Такое разделение позволяет исключить из выпуклой оптимизационной постановки дискретный выбор стороны обхода

препятствия и сохранить линейную форму ограничений свободного пространства.

Формирование опорного пути, не пересекающего препятствия, методом A^*

Перед формированием полигона проходимости выполняется поиск связного опорного пути, не пересекающего границы проезжей части и статические препятствия. Данный этап определяет допустимый вариант прохождения свободного пространства, в том числе сторону обхода препятствий. Необходимость предварительного поиска обусловлена тем, что исходная опорная линия может пересекать занятые области, а свободное пространство в пределах рассматриваемого участка может состоять из нескольких несвязных областей.

Выбор одного связного прохода должен быть выполнен до решения задачи квадратичного программирования. Объединение свободных областей, расположенных по разные стороны препятствия, не может быть представлено одной системой интервальных линейных ограничений без введения дополнительных дискретных переменных. Поэтому на первом этапе решается дискретная задача поиска пути, а на последующем этапе выполняется непрерывная оптимизация его геометрических характеристик.

Исходная опорная линия задается дискретной последовательностью точек и аппроксимируется кубическим B -сплайном. На полученной кривой с равномерным шагом по дуговой координате выбираются точки $\mathbf{r}_0(s_i)$. Для каждой точки определяется единичный вектор нормали к опорной линии:

$$\mathbf{n}_0(s_i) = \begin{bmatrix} -\sin \theta_{0,i} \\ \cos \theta_{0,i} \end{bmatrix}, \quad (3.28)$$

где $\theta_{0,i}$ — угол касательной к опорной линии в точке, соответствующей дуговой координате s_i .

В каждом продольном сечении формируется набор точек, смещенных относительно опорной линии в поперечном направлении:

$$\mathbf{p}_{i,j} = \mathbf{r}_0(s_i) + d_j \mathbf{n}_0(s_i), \quad d_j = j \Delta d, \quad (3.29)$$

где $i = 0, \dots, N$ — номер продольного слоя графа; $j = -M, \dots, M$ — номер

поперечного положения; Δd — шаг поперечной дискретизации; d_j — поперечное смещение относительно исходной опорной линии.

Максимальное рассматриваемое поперечное смещение определяется выражением

$$d_{\max} = M\Delta d. \quad (3.30)$$

Каждой допустимой точке $\mathbf{p}_{i,j}$ ставится в соответствие вершина $v_{i,j}$ ориентированного графа $G = (V, E)$. Вершины соседних продольных слоев соединяются ребрами. Множество возможных переходов из вершины $v_{i,j}$ задается условием

$$\mathcal{K}(j) = \{k \in \{-M, \dots, M\} : |k - j| \leq m_c\}, \quad (3.31)$$

где m_c — наибольшее допустимое изменение номера поперечного положения за один продольный шаг. Ограничение (3.31) исключает переходы с чрезмерным изменением поперечного положения и уменьшает число ребер графа.

Для оценки близости вершины к занятым областям используется карта расстояний. Расстояние от точки графа до ближайшей занятой области определяется соотношением

$$D_{i,j} = D(\mathbf{p}_{i,j}) = \min_{\mathbf{z} \in \Omega_{\text{зан}}} \|\mathbf{p}_{i,j} - \mathbf{z}\|_2, \quad (3.32)$$

где $\Omega_{\text{зан}}$ — множество точек, принадлежащих статическим препятствиям и областям за границами проезжей части. Если карта расстояний содержит значения, выраженные в числе ячеек, они предварительно переводятся в метры с учетом пространственного разрешения карты.

Вершина включается в граф только при выполнении условия

$$v_{i,j} \in V \iff D_{i,j} \geq D_{\min}, \quad (3.33)$$

где $D_{\min}$ — минимально допустимое расстояние до занятой области на этапе графового поиска.

Ребро между двумя вершинами соседних продольных слоев включается в граф, если его конечная вершина принадлежит множеству допустимых переходов

и отрезок между вершинами не пересекает занятую область:

$$(v_{i,j}, v_{i+1,k}) \in E \iff k \in \mathcal{K}(j),$$

$$\min_{\lambda \in [0,1]} D((1-\lambda)\mathbf{p}_{i,j} + \lambda\mathbf{p}_{i+1,k}) \geq D_{\min}. \quad (3.34)$$

При численной реализации условие (3.34) проверяется в конечном числе промежуточных точек, расположенных вдоль ребра с шагом, не превышающим размер ячейки карты.

Для каждой допустимой вершины назначается неотрицательная стоимость, учитывающая степень следования исходной опорной линии и удаленность от занятых областей:

$$q_{i,j} = W_g \left(\frac{d_j}{d_{\max}} \right)^2 + W_o \Phi(D_{i,j}), \quad (3.35)$$

где W_g — весовой коэффициент следования исходной опорной линии; W_o — весовой коэффициент удаления от препятствий.

Штраф за приближение к занятой области задается кусочно-квадратичной функцией:

$$\Phi(D) = \begin{cases} \left(\frac{D_0 - D}{D_0 - D_{\min}} \right)^2, & D_{\min} \leq D < D_0, \\ 0, & D \geq D_0, \end{cases} \quad (3.36)$$

где $D_0 > D_{\min}$ — расстояние, ограничивающее область влияния препятствия.

При $D = D_{\min}$ значение функции (3.36) равно единице и соответствует наибольшему штрафу для допустимой вершины. По мере удаления от препятствия штраф уменьшается, а при $D \geq D_0$ принимается равным нулю. Точки, для которых $D < D_{\min}$, не оцениваются, поскольку соответствующие им вершины исключаются из графа в соответствии с условием (3.33).

Применение нормированных слагаемых в выражении (3.35) позволяет сопоставлять их значения независимо от ширины области поперечного поиска и выбранного расстояния влияния препятствий. Отношение W_o/W_g определяет соотношение между предпочтением движения вблизи исходной опорной линии и предпочтением движения с увеличенным расстоянием до занятых областей.

Стоимость перехода между вершинами соседних продольных слоев

принимается равной стоимости конечной вершины:

$$c(v_{i,j}, v_{i+1,k}) = q_{i+1,k}. \quad (3.37)$$

Поскольку каждый полный путь содержит по одной вершине каждого продольного слоя, выражение (3.37) обеспечивает сопоставимую оценку всех рассматриваемых вариантов прохождения.

При поиске методом A^* для каждой вершины вычисляются накопленная стоимость пути от начальной вершины и оценка стоимости оставшейся части пути. Накопленная стоимость определяется выражением

$$G(v_{i,j}) = \sum_{(v_{\ell,m}, v_{\ell+1,n}) \in \pi_{0 \rightarrow (i,j)}} c(v_{\ell,m}, v_{\ell+1,n}), \quad (3.38)$$

где $\pi_{0 \rightarrow (i,j)}$ — путь от начальной вершины до вершины $v_{i,j}$.

Для оценки наименьшей возможной стоимости оставшейся части пути учитывается ограничение на изменение поперечного положения. Множество номеров поперечных положений, достижимых в слое ℓ из вершины $v_{i,j}$ без учета препятствий, определяется как

$$\mathcal{I}_{\ell}(i, j) = \{m \in \{-M, \dots, M\} : |m - j| \leq m_c(\ell - i)\}. \quad (3.39)$$

Эвристическая оценка оставшейся стоимости задается выражением

$$H(v_{i,j}) = \sum_{\ell=i+1}^N \min_{m \in \mathcal{I}_{\ell}(i,j)} \left[W_g \left(\frac{d_m}{d_{\max}} \right)^2 \right]. \quad (3.40)$$

В выражении (3.40) не учитывается неотрицательный штраф за приближение к препятствиям, поэтому полученная величина не превышает действительную стоимость оставшейся части пути. Следовательно, данная оценка является допустимой для применения в алгоритме A^* и не требует введения дополнительного весового коэффициента.

Для каждой рассматриваемой вершины вычисляется суммарная оценка

$$F(v_{i,j}) = G(v_{i,j}) + H(v_{i,j}). \quad (3.41)$$

На очередном шаге поиска из множества еще не обработанных вершин

выбирается вершина с наименьшим значением $F(v_{i,j})$. После достижения конечной вершины последовательность вершин пути восстанавливается по сохраненным связям с предшествующими вершинами.

В результате определяется путь минимальной суммарной стоимости:

$$\pi^* = \arg \min_{\pi \in \Pi(v_0, v_N)} \sum_{(u,v) \in \pi} c(u, v), \quad (3.42)$$

где $\Pi(v_0, v_N)$ — множество допустимых путей между начальной v_0 и конечной v_N вершинами рассматриваемого участка.

Эвристическая оценка (3.40) используется только для определения порядка рассмотрения вершин и не изменяет целевой функционал (3.42). Таким образом, стоимость пути содержит только два заданных слагаемых: штраф за отклонение от исходной опорной линии и штраф за приближение к препятствиям.

Результат поиска опорного пути методом A^* представлен на рисунке 27.

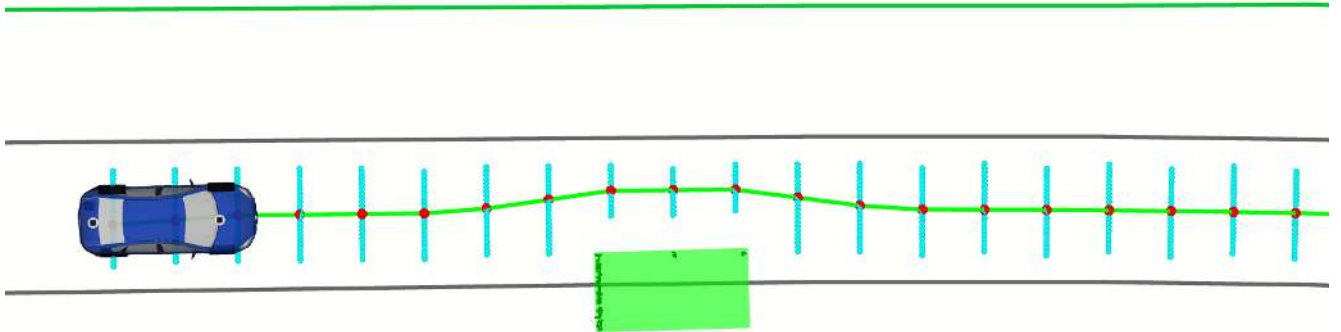


Рисунок 27 – Результат формирования опорного пути, не пересекающего статическое препятствие, методом A^*

На рисунке 27 голубыми точками показаны вершины графа, полученные путем поперечного смещения точек исходной опорной линии. Красными точками обозначена последовательность вершин найденного пути, а зеленой линией — соединяющие их ребра. Серые линии соответствуют границам проезжей части, полупрозрачная прямоугольная область — статическому

препятствию. Ребра графа, не вошедшие в найденный путь, на рисунке не показаны.

На участке расположения препятствия найденный путь смещается относительно исходной опорной линии и определяет выбранную сторону обхода. На участках, свободных от препятствий, первое слагаемое выражения (3.35) обеспечивает возвращение пути к исходной опорной линии.

Найденный путь имеет кусочно-линейный характер, обусловленный дискретным представлением графа, и поэтому не используется в качестве окончательной траектории автомобиля. Последовательность найденных вершин подвергается сглаживанию. Полученная кривая используется как опорный путь для построения системы координат Френе и формирования полигона проходимости.

При построении полигона проходимости в каждом продольном сечении выбирается связный интервал свободного пространства, содержащий соответствующую точку сглаженного пути. Последовательности левых и правых границ выбранных интервалов образуют границы полигона проходимости. Такой порядок построения исключает объединение свободных областей, расположенных по разные стороны препятствия, и сохраняет вариант обхода, найденный методом A^* .

Таким образом, поиск на графе решает дискретную задачу выбора связного варианта прохождения между границами проезжей части и статическими препятствиями. Последующая квадратичная оптимизация выполняется внутри выбранного полигона проходимости и обеспечивает сглаживание траектории, ограничение кривизны и скорости ее изменения, а также учет габаритов транспортного средства.

Определение модели габаритов и проверка коллизий

Формирование полигона проходимости выполняется по схеме, приведенной на рисунке 28.

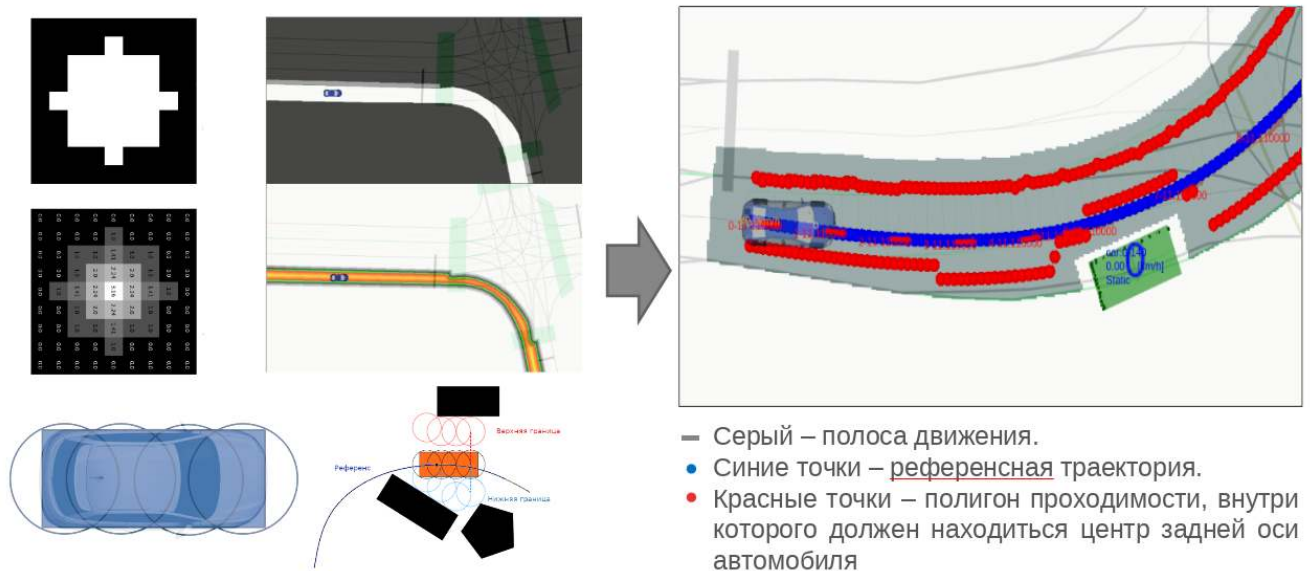


Рисунок 28 – Формирование полигона проходимости с учетом габаритов автомобиля

На рисунке 28 показана последовательность формирования полигона проходимости: исходное представление занятых областей; аппроксимация транспортного средства покрывающими окружностями; проверка допустимости положения окружностей; формирование множества допустимых точек; построение верхней и нижней границ допустимого коридора

Геометрия транспортного средства аппроксимируется четырьмя перекрывающимися окружностями, центры которых расположены вдоль продольной оси автомобиля (рисунок 29). Указанная аппроксимация обеспечивает вычислительно эффективную проверку допустимости положения транспортного средства относительно препятствий и границ проезжей части.

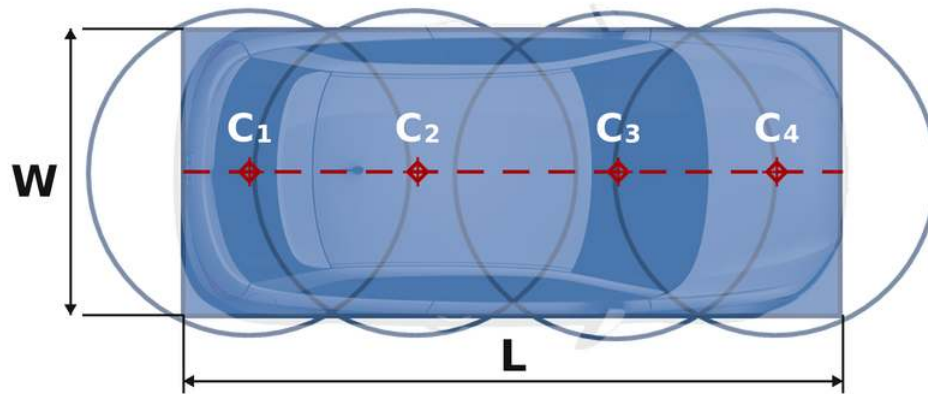


Рисунок 29 – Изображение транспортного средства в четырех кругах

Здесь L — длина автомобиля, W — ширина автомобиля, $d_1, \dots, d_4$ — продольные смещения центров «контрольных» окружностей относительно оси задних колес. Радиус окружностей вычисляется по формуле:

$$R_{circle} = \sqrt{(L/8)^2 + (W/2)^2} \quad (3.43)$$

Проверка на столкновение выполняется путем сопоставления расстояний от центров окружностей до ближайших границ допустимого коридора с величиной радиуса R_{circle} . На основе карты препятствий и дорожной разметки формируется сетка дискретных точек в координатах (s, d) . Для каждой точки сетки выполняется проверка коллизий с учетом всех окружностей аппроксимации; точки, для которых хотя бы одна окружность пересекается с препятствием либо выходит за границы дороги, исключаются из допустимого множества.

Граница полученного множества допустимых точек формирует полигон проходимости, который задает верхнюю и нижнюю границы допустимого поперечного смещения как функции продольной координаты s . Указанные границы далее используются при формировании ограничений неравенства в задаче квадратичной оптимизации траектории.

Для каждого дискретного состояния опорного пути P_k центры окружностей $c_0 \dots c_3$ вычисляются как точки на продольной оси кузова, смещенные на d_i :

$$c_i = (x + d_i \cos \theta, y + d_i \sin \theta) \quad (3.44)$$

Получение границ коридора через карту дистанций

Границы коридора $[d_{k,i}^L, d_{k,i}^R]$ в формуле 3.47 определяются на основе карты дистанций, позволяющей в каждой ячейке дискретного пространства хранить расстояние до ближайшей занятой области. К занятым относятся ячейки, соответствующие статическим препятствиям, а также недопустимым зонам за границами проезжей части. Такое представление позволяет рассматривать границы дороги и препятствия в рамках единой расчетной процедуры без отдельного определения расстояний до каждого геометрического объекта.

Исходными данными для построения карты дистанций является бинарная карта занятости, в которой каждой ячейке ставится в соответствие признак свободного или занятого пространства. Карта дистанций формируется путем дистанционного преобразования бинарной карты и последующего пересчета полученных значений в метрические величины с учетом пространственного разрешения сетки. В результате для каждой свободной ячейки определяется минимальное расстояние до множества занятых ячеек.

В общем виде расстояние в точке (ячейке) p можно определить как:

$$D(p) = \min_{q \in \Omega_{occ}} \|p - q\|_2,$$

где Ω_{occ} — множество занятых ячеек. Значение $D(p)$ используется для вычисления допустимых боковых зазоров (слева и справа) относительно локальной нормали к опорной линии в узле s_k для каждой контрольной окружности. Обеспечивается получение двух границ поперечного смещения, задающих допустимый интервал перемещения окружности без пересечения с занятыми областями (см. Приложение А).

Иллюстрация принципа формирования карты дистанций приведена на рисунке 30 а); пример карты дистанций, сформированной для разработанного алгоритма, представлен на рисунке 30 б).

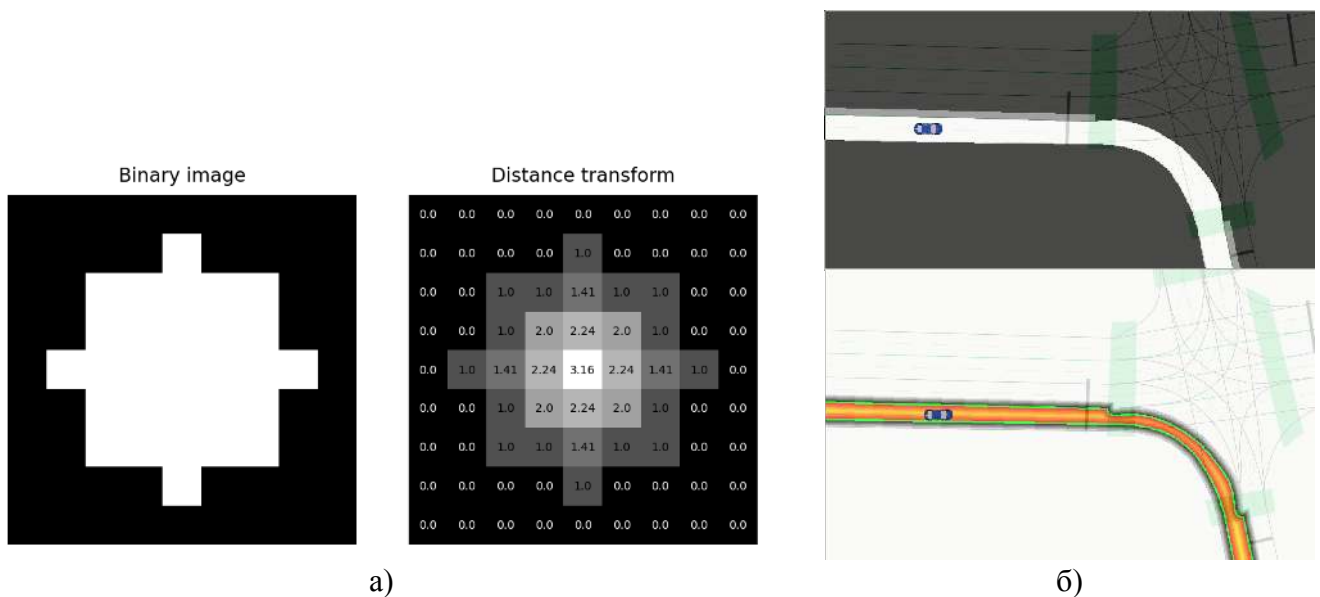


Рисунок 30 – а) двоичная карта занятости и соответствующая карта расстояний до ближайшей занятой ячейки; белые ячейки соответствуют свободному пространству, черные — занятому; б) сопоставление карты расстояний с обычной картой занятости на примере дорожной сцены

Для целей контроля корректности построения коридора границы допустимой области движения дополнительно визуализируются в декартовой системе координат в виде набора граничных точек, формирующих контур допустимого пространства (рисунок 31). Указанная визуализация носит иллюстративный характер и не изменяет постановку оптимизации, сформулированную в координатах Френе.

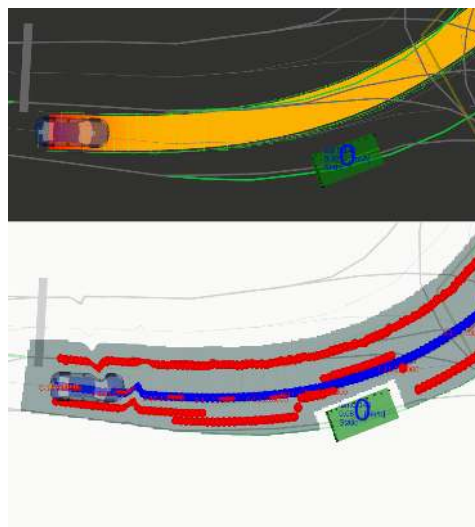


Рисунок 31 – Визуализация многоугольника допустимого пространства, отображенного в декартовом пространстве (красные точки)

Граничные точки многоугольника допустимого пространства определяются по карте дистанций на основе вычисления расстояний до ближайших занятых областей и последующего построения границ допустимого коридора. Алгоритм построения многоугольника представлен в Приложении 2. Визуализация предназначена для контроля корректности геометрических ограничений и не изменяет постановку оптимизационной задачи в координатах Френе.

Преобразование геометрии в линейные ограничения QR

В соответствии с разработанным методом QR-постановка задана в переменных оптимизации (см. формулу 3.19), включающих поперечное отклонение d , угловую ошибку e_θ и кривизну k . Для учета габаритов задается соответствие поперечного положения каждой контрольной окружности с переменными d и e_θ .

Для малых значений угловой ошибки e_θ поперечное смещение центра i -й окружности аппроксимируется линейным выражением:

$$y_{circle,i} \approx d + d_i e_\theta \quad (3.45)$$

Вдоль оси s соответствующая продольная позиция центра окружности выражается как:

$$s_{circle,i} \approx s_k + d_i \quad (3.46)$$

Пусть для каждого узла k и окружности i предварительно вычислены допустимые границы коридора $[d_{k,i}^L, d_{k,i}^R]$ (метод их получения приведен в подпункте 3.3.2.3). Тогда геометрические ограничения записываются в виде интервального неравенства:

$$d_{C_i} \in [d_{k,i}^L, d_{k,i}^R] \quad (3.47)$$

При этом продольный сдвиг d_i не вводится в QR явно. Ограничения коридора заранее вычисляются для каждой окружности на ее собственной продольной позиции $s_k + d_i$, поэтому внутри одного шага QR используется только поперечная линейная связь $d_{C_i} = d + d_i e_\theta$. В результате ограничения по габаритам сохраняют линейный вид относительно переменных d и e_θ , что обеспечивает сохранение выпуклости QR-постановки и исключает

необходимость включения в ограничения нелинейных функций вида $\sin(\cdot)$ и $\cos(\cdot)$. При необходимости допускается мягкое соблюдение ограничений путем введения переменных слабины со штрафованием в целевой функции (см. раздел 3.3.3).

3.3.3 Формирование функционала качества

Формирование функционала качества производится по целевой функции, объединяющей требования к точности следования опорной линии, ограничению кривизны и скорости ее изменения, а также к мягкому соблюдению ограничений безопасности. В используемой постановке целевая функция задачи оптимизации задается в следующем виде:

$$J = \sum_{i=0}^{N-1} (W_d d_i^2 + W_c k_i^2 + W_{safe} \epsilon_{safe,i}^2) + \sum_{i=0}^{N-2} W_{cr} \left(\frac{k_{i+1} - k_i}{\Delta s} \right)^2. \quad (3.48)$$

где W_d — весовой коэффициент, задающий степень штрафа за поперечное отклонение от опорной траектории (переменная d); W_c — весовой коэффициент, задающий степень штрафа за величину кривизны k ; W_{cr} — весовой коэффициент, задающий степень штрафа за скорость изменения кривизны k' ; W_{safe} — весовой коэффициент на переменные слабины $\epsilon_{safe,i}$, используемые для мягкого соблюдения ограничений вблизи препятствий и границ проезжей части.

Таким образом, сформированный функционал задает компромисс между точностью следования, плавностью траектории и требованиями безопасности. Далее в подразделе 3.3.4 приводятся результаты моделирования и анализ влияния выбранных весовых коэффициентов.

3.3.4 Результаты моделирования

В данном подразделе приведены результаты моделирования, подтверждающие работоспособность постановки сглаживания траектории в системе координат Френе при наличии линейных ограничений на свободное пространство (границы проезжей части) и кинематических ограничений транспортного средства.

Условия вычислительного эксперимента. Опорная линия движения задается дискретным набором точек и предварительно аппроксимируется (см. принятый формат опорной траектории и соответствующие определения в предыдущих подразделах). Оптимизация выполняется в координатах Френе на горизонте из $N+1$ узлов при дискретизации по дуговой координате с шагом Δs ; используемые значения $\{N, \Delta s\}$ выбираются из условия применения в реальном времени и согласуются с оценкой вычислительных затрат в подразделе, посвященном ускорению решения QR. Границы проезжей части задаются в виде интервальных ограничений $d_{min,i} \leq d_i \leq d_{max,i}$, сформированных по карте дистанций, а учет габаритов транспортного средства выполняется за счет переноса границ к покрывающим окружностям (см. подраздел об учете геометрии автомобиля). Для обеспечения выполнимости при узком коридоре допускается мягкое соблюдение ограничений посредством переменных слабины $\epsilon_{safe,i}$, входящих в функционал качества (см. подраздел 3.3.3).

На рисунке 32 показан результат сглаживания траектории с явным учетом границ полосы движения. В сравнении с вариантом без ограничений по свободному пространству наблюдается увеличение пиковых значений кривизны $|k|$ и производной кривизны $|k'|$ (по дуговой координате), что является ожидаемым следствием ограниченности допустимого коридора.

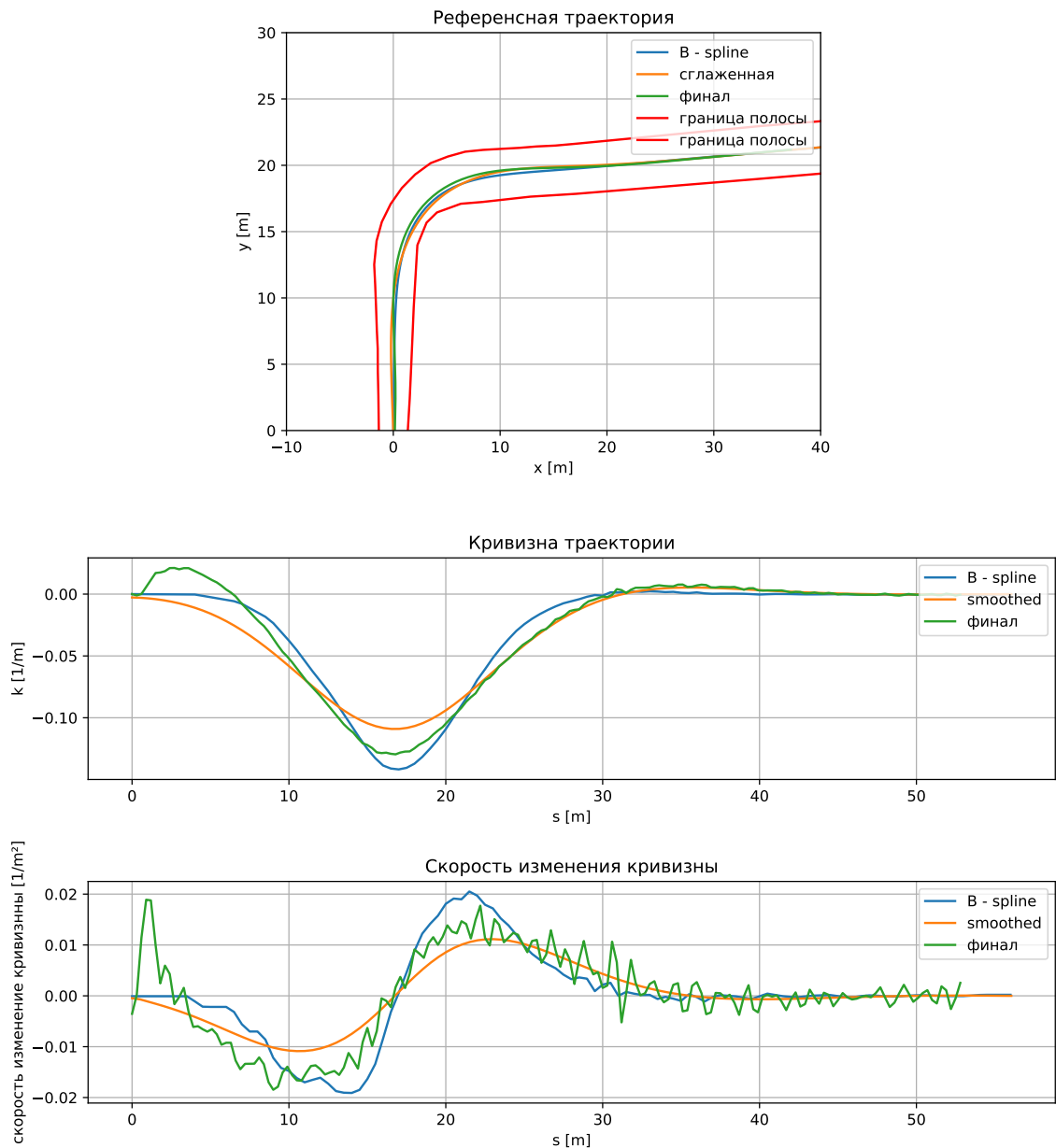


Рисунок 32 – Сглаживание траектории вхождения в поворот с учетом полосы движения ($W_d = 0.1$; $W_c = 10.0$; $W_{cr} = 200.0$)

Ключевым результатом является выполнение геометрической допустимости: траектория формируется с зазором до бордюра/границ и не пересекает недопустимые области.

Однако в режиме резкого поворота траектория на рисунке 32 располагается близко к внутренней границе. Для повышения робастности к ошибкам исполнения (погрешности слежения и локализации) введена модификация функционала, обеспечивающая предпочтение движения по *внешней* стороне поворота.

Модификация функционала для смещения к внешней стороне поворота.
 Исходный функционал J , заданный в подразделе 3.3.3, дополняется добавочным

членом

$$J_{turn} = \sum_{i=0}^{N-1} W_{d,turn} \chi_i (d_i - d_{turn,i})^2, \quad (3.49)$$

где $W_{d,turn}$ — весовой коэффициент смещения (при этом должно выполняться $W_{d,turn} > W_d$), а индикатор включения смещения определяется порогом по кривизне опорной линии:

$$\chi_i = \begin{cases} 1, & |k_{ref,i}| \geq k_{turn}, \\ 0, & |k_{ref,i}| < k_{turn}. \end{cases} \quad (3.50)$$

Целевое поперечное положение $d_{turn,i}$ задается как смещение от середины коридора к *внешней* границе поворота:

$$d_{turn,i} = (1 - l_{turn}) d_{c,i} + l_{turn} d_{out,i}, \quad d_{c,i} = \frac{d_{min,i} + d_{max,i}}{2}, \quad (3.51)$$

$$d_{out,i} = \begin{cases} d_{min,i}, & k_{ref,i} \geq k_{turn} \quad (\text{поворот влево, внешняя граница справа}), \\ d_{max,i}, & k_{ref,i} \leq -k_{turn} \quad (\text{поворот вправо, внешняя граница слева}), \\ d_{c,i}, & |k_{ref,i}| < k_{turn}, \end{cases} \quad (3.52)$$

где параметр $l_{turn} \in [0, 1]$ задает степень смещения к внешней границе: при $l_{turn} = 0$ целевое положение совпадает с серединой коридора, при $l_{turn} = 1$ целевое положение совпадает с внешней границей (с учетом габаритов транспортного средства, если границы перенесены к покрывающим окружностям).

Итоговый функционал принимает вид $\tilde{J} = J + J_{turn}$ и сохраняет квадратичную структуру, что позволяет решать задачу в рамках QP без изменения класса оптимизации. Пример траектории сгенерированной с данным функционалом представлен на рисунке 33.

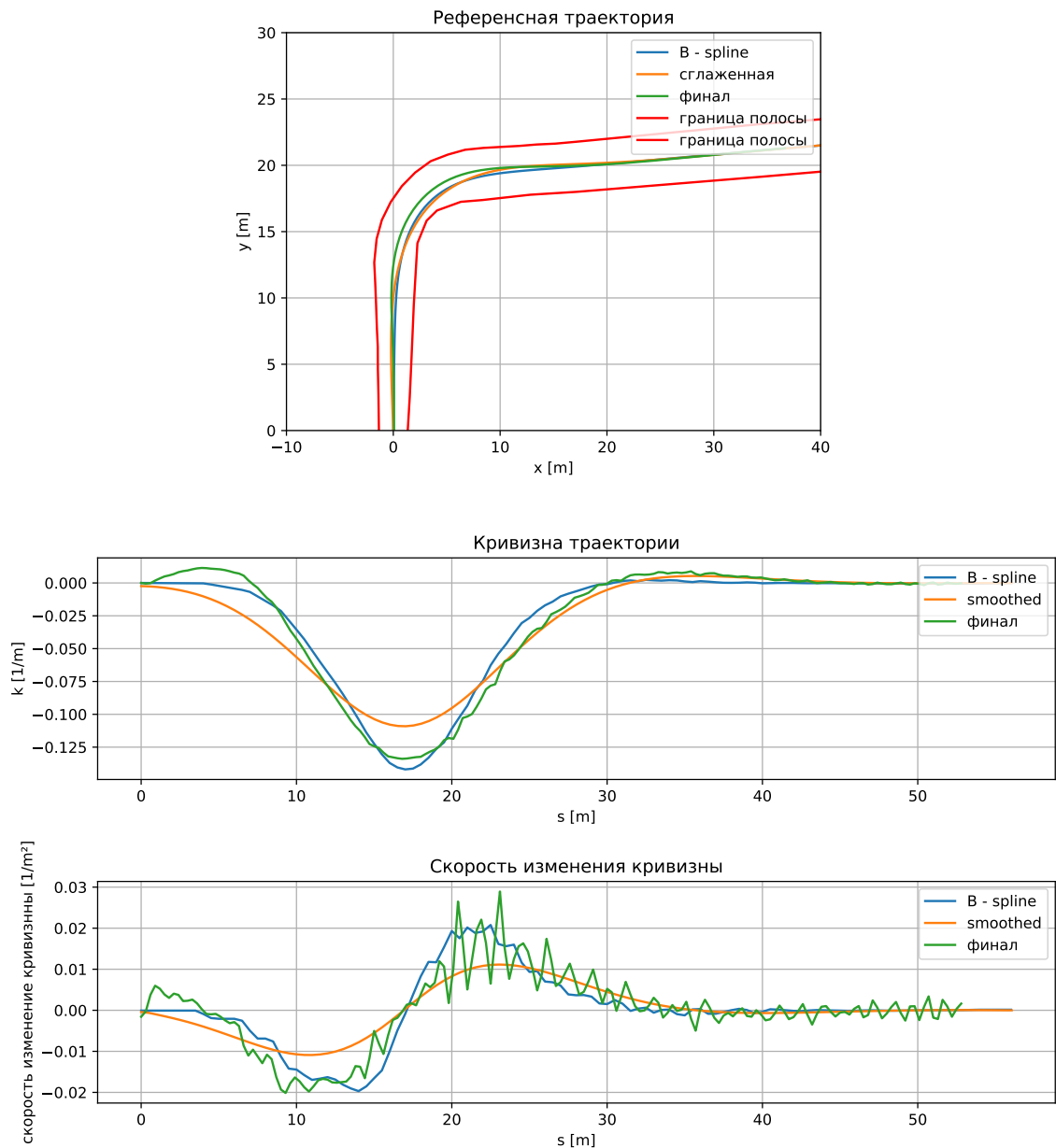


Рисунок 33 – Сглаживание траектории вхождения в поворот по внешней стороне с учетом полосы движения
 $(W_d = 0.1; W_c = 10.0; W_{cr} = 200.0; W_{d,turn} = 0.8; l_{turn} = 0.8; k_{turn} = 0.09)$

Из рисунка 33 следует, что введение J_{turn} обеспечивает смещение траектории к внешней стороне поворота, то есть увеличение радиуса вхождения при сохранении соблюдения границ полосы. При этом возможен дополнительный рост показателей, характеризующих резкость по кривизне (в частности, $|k'|$), что рассматривается как допустимый компромисс в пользу запаса безопасности относительно бордюра при наличии ошибок исполнения.

Далее алгоритм протестирован на различных геометриях дороги.

На рисунке 34 представлено прохождение S-образной траектории.

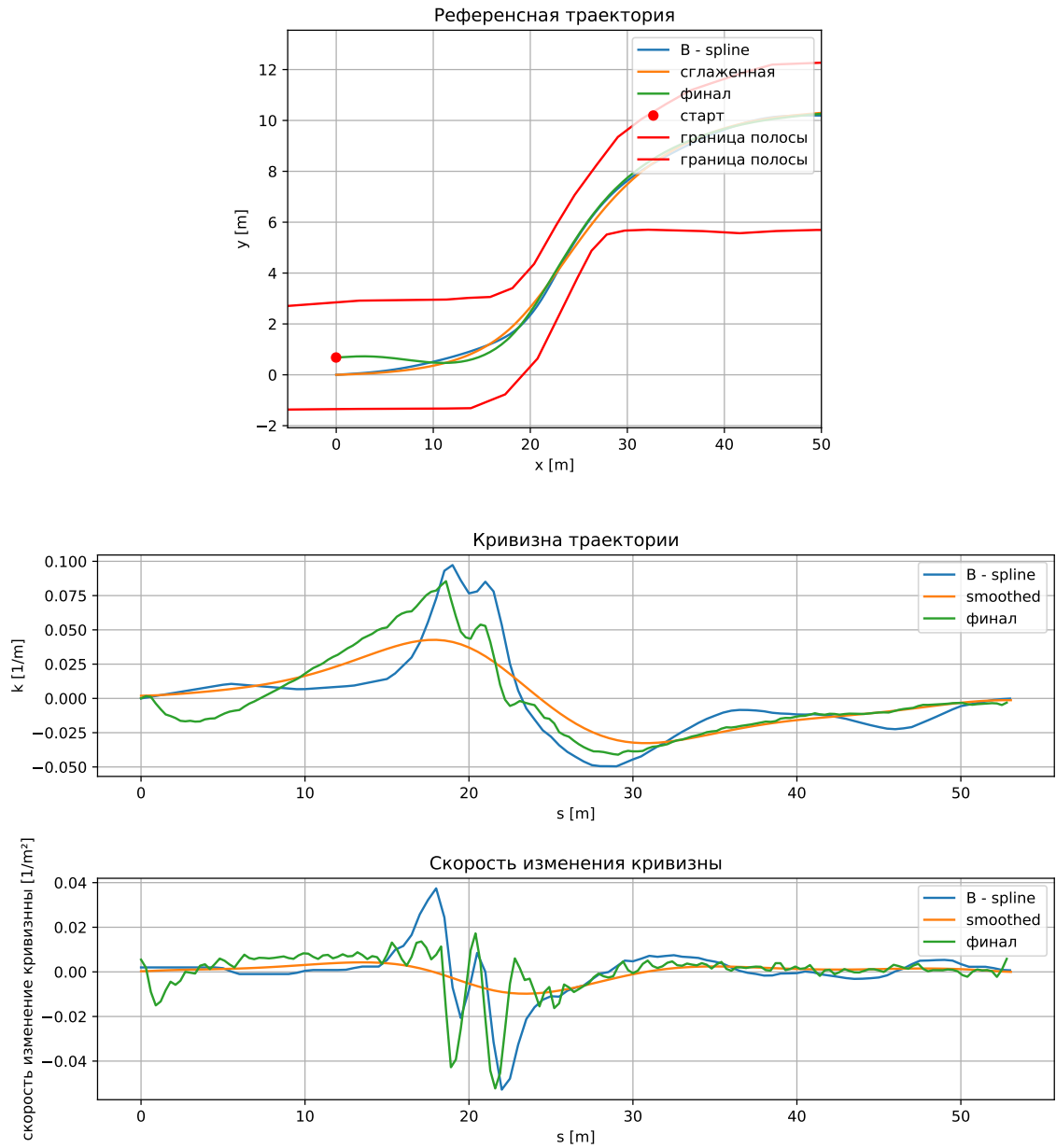


Рисунок 34 – Сглаживание траектории с учетом полосы движения на S-образной геометрии
 $(W_d = 0.1; W_c = 10.0; W_{cr} = 200.0; W_{d,turn} = 0.8; l_{turn} = 0.8; k_{turn} = 0.09)$

На рисунке 35 представлено прохождение траектории с несколькими поворотами (изменяющееся направление кривизны).

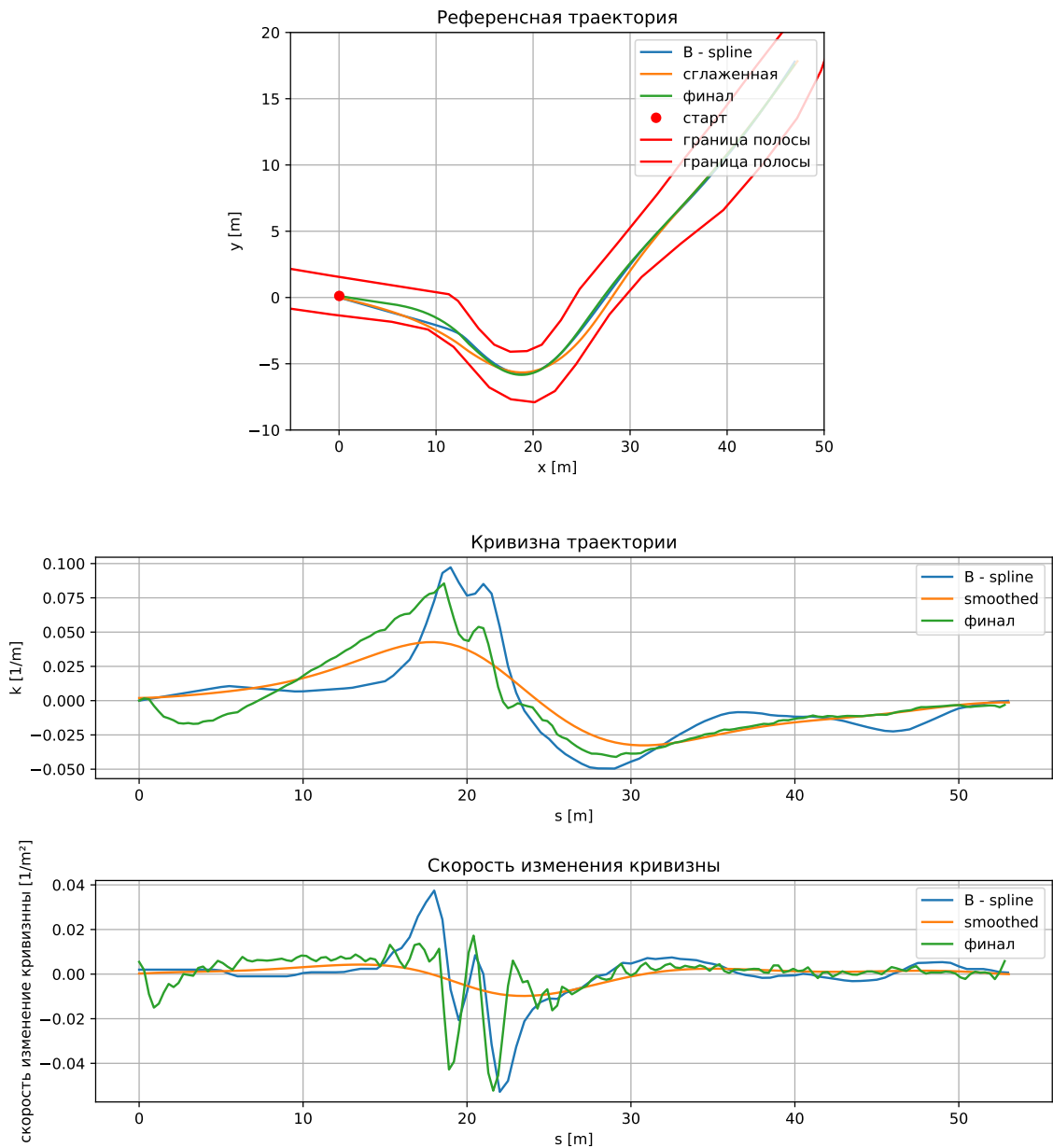


Рисунок 35 – Сглаживание траектории с учетом полосы движения на траектории с несколькими поворотами
 $(W_d = 0.1; W_c = 10.0; W_{cr} = 200.0; W_{d,turn} = 0.8; l_{turn} = 0.8; k_{turn} = 0.09)$

По результатам, представленным на рисунках 34 и 35, установлено, что предложенная постановка формирует траектории удержания в полосе движения в различных дорожных условиях при соблюдении ограничений по свободному пространству.

В качестве метрик используются: минимальные зазоры до границ коридора Δ_{min}^L и Δ_{min}^R , пиковые значения $\max |k|$ и $\max |k'|$, показатели использования мягких ограничений $\max \epsilon_{safe}$ и долю узлов, в которых $\epsilon_{safe,i} > 0$ (см. таблица 11).

Таблица 11 – Сводные метрики сглаживания траектории в координатах Френе

Сценарий	Конфигурация	Δ_{min}^L , м	Δ_{min}^R , м	$\max k $, 1/м	$\max k' $, 1/м ²	$\max \epsilon_{safe}$
Рис. 3.29	без J_{turn}	≈ 0.25	≈ 1.80	≈ 0.125	≈ 0.019	≈ 0
Рис. 3.30	с J_{turn}	≈ 0.80	≈ 0.90	≈ 0.135	≈ 0.028	≈ 0
Рис. 3.31	с J_{turn}	≈ 0.60	≈ 0.60	≈ 0.110	≈ 0.026	≈ 0
Рис. 3.32	с J_{turn}	≈ 0.50	≈ 0.60	≈ 0.120	≈ 0.030	≈ 0

По данным таблицы 11 установлено, что введение ограничений по свободному пространству обеспечивает соблюдение границ коридора движения и формирует положительный минимальный зазор до границ Δ_{min}^L и Δ_{min}^R во всех рассмотренных сценариях. При этом модификация функционала с добавочным членом J_{turn} приводит к перераспределению зазоров в пользу внешней стороны поворота (увеличение минимального зазора со стороны внутренней границы), что повышает робастность к ошибкам исполнения траектории. Одновременно наблюдается рост пиковых значений $\max |k|$ и $\max |k'|$, что отражает компромисс между плавностью траектории и обеспечением геометрической допустимости в ограниченном коридоре. В рассматриваемых сценариях значения $\max \epsilon_{safe}$ близки к нулю, что свидетельствует о выполнении ограничений без существенного задействования мягких переменных.

Замечание о скорости движения. В настоящем подразделе рассматривается геометрическое планирование траектории; динамические ограничения по скорости напрямую в постановку не включаются. В режиме следования траектории допустимая кривизна ограничивается, в первую очередь, предельным углом поворота управляемых колес и геометрией коридора. Обеспечение динамической реализуемости по боковым ускорениям и комфорту достигается на следующем этапе за счет формирования скоростного профиля, снижающего скорость на участках повышенной кривизны.

3.3.5 Формирование коридора движения с отдельным учетом границ дороги и статических препятствий

Для отдельного учета статических препятствий в постановку геометрического планирования введена модификация, заключающаяся в раздельном задании защитных зазоров относительно границ дороги (границ полосы, бордюра) и статических препятствий (например, припаркованных транспортных средств). Указанное разделение обусловлено тем, что требования к запасу безопасности относительно препятствий, как правило, являются более строгими, чем требования к запасу относительно геометрических границ проезжей части, особенно в условиях ограниченной ширины полосы движения.

Пусть в системе координат Френе для каждой дискретной точки горизонта s_i по данным карты/разметки сформирован допустимый интервал поперечного смещения, определяемый только геометрией дороги, $[d_{\min,i}^{\text{road}}, d_{\max,i}^{\text{road}}]$, а по данным подсистемы восприятия сформированы ограничения, обусловленные статическими препятствиями, $[d_{\min,i}^{\text{obs}}, d_{\max,i}^{\text{obs}}]$, рассчитанные с учетом габаритов транспортного средства. Вводятся два независимых параметра защитного зазора: Δ_{road} для границ дороги и Δ_{obs} для препятствий, причем обычно выбирается $\Delta_{\text{obs}} > \Delta_{\text{road}}$. Итоговые границы коридора движения для задачи QR вычисляются как пересечение указанных ограничений с учетом соответствующих зазоров и задаются линейными неравенствами в виде

$$\begin{aligned} d_{\min,i} &= \max(d_{\min,i}^{\text{road}} + \Delta_{\text{road}}, d_{\min,i}^{\text{obs}} + \Delta_{\text{obs}}), \\ d_{\max,i} &= \min(d_{\max,i}^{\text{road}} - \Delta_{\text{road}}, d_{\max,i}^{\text{obs}} - \Delta_{\text{obs}}). \end{aligned} \quad (3.53)$$

что обеспечивает различную “жесткость” требований к расстоянию до границ дороги и до препятствий при сохранении единого формата ограничений $d_{\min,i} \leq d_i \leq d_{\max,i}$. В случае, когда препятствия формируют несколько допустимых интервалов по d для одной и той же точки s_i , в постановке выбирается интервал, согласованный с опорным положением (например, содержащий $d_{\text{ref},i}$ либо минимизирующий отклонение $|d - d_{\text{ref},i}|$), что снижает вероятность скачкообразного изменения решения на соседних шагах дискретизации.

Тестирование алгоритма выполнено с визуализацией в среде RViz [73]. На рисунке 36 приведена последовательность состояний при обнаружении статического препятствия на прямолинейном участке. На подрисунке (а)

препятствия отсутствуют, и траектория формируется в пределах полосы движения относительно опорной линии. На подрисунке (б) показана визуализация свободного пространства в пределах полосы, полученного по границам дороги и данным о препятствиях с учетом габаритов транспортного средства. На подрисунке (в) показан результат пересчета коридора и решения задачи QR: траектория объезда остается в пределах полосы и обеспечивает отдельные зазоры Δ_{road} и Δ_{obs} до границ дороги и до препятствия соответственно. Следует отметить, что в рамках данного подраздела рассматривается геометрическое планирование; динамические ограничения, зависящие от профиля скорости (например, по боковым ускорениям), непосредственно в постановку не включаются и обеспечиваются на последующем этапе за счет формирования скоростного профиля.

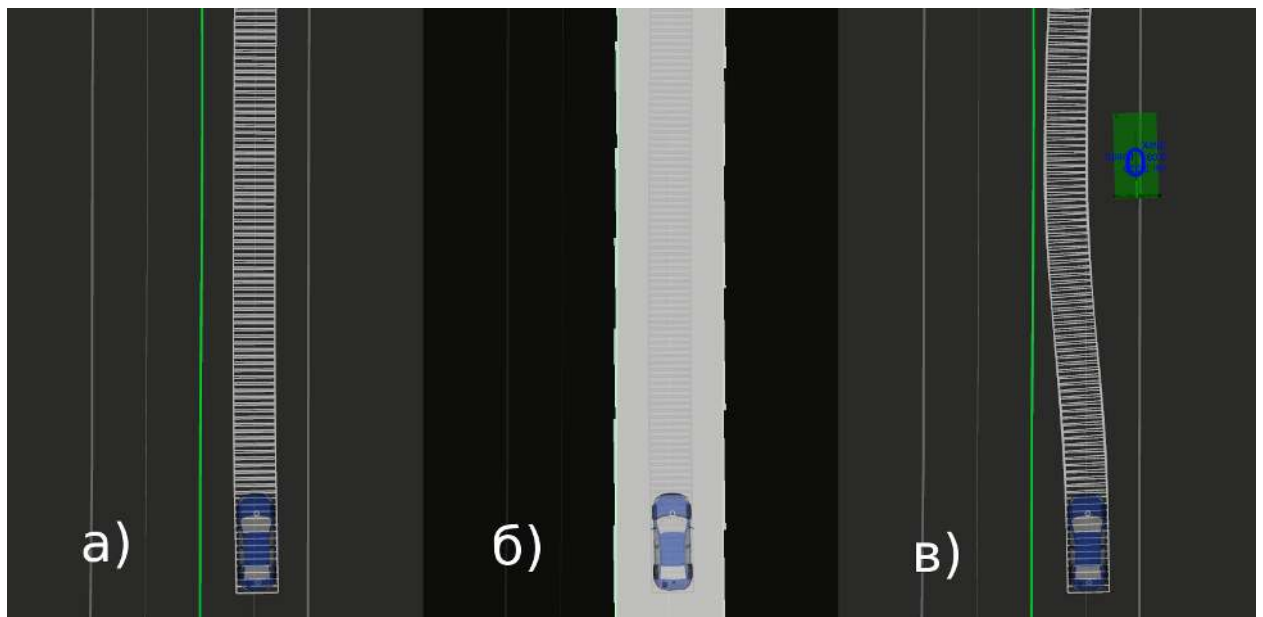


Рисунок 36 – Объезд препятствия на прямой. а) Препятствие не обнаружено, б) визуализируется свободное пространство на полосе движения, в) объезд препятствия на полосе движения

По результатам, представленным на рисунке 36, подтверждается возможность формирования плавной траектории объезда в пределах полосы движения при появлении статического препятствия. Получаемая траектория является допустимой относительно сформированного коридора и обеспечивает отдельные защитные зазоры до границ дороги и до препятствия, задаваемые параметрами Δ_{road} и Δ_{obs} соответственно.

Рисунок 37 иллюстрирует сценарий, в котором в процессе геометрического планирования в ограниченном пространстве может формироваться траектория с самопересечением в декартовом пространстве. Данный эффект рассматривается как допустимый для маневров разворота и смены направления движения при условии, что траектория удовлетворяет ограничениям допустимой области движения и не пересекает запрещенные области, сформированные по границам дороги и статическим препятствиям.

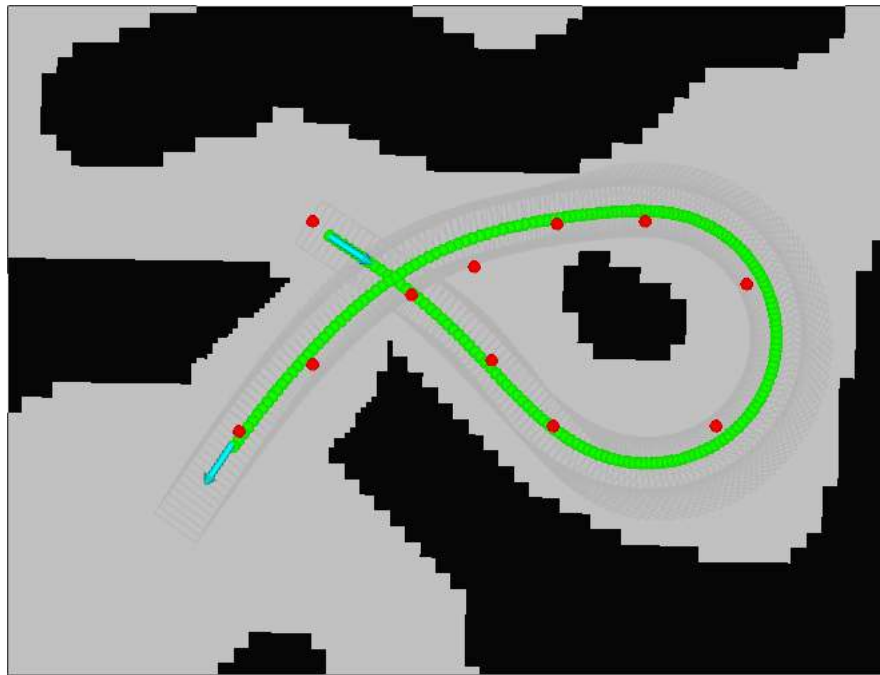


Рисунок 37 – Пример генерации самопересекающейся траектории

Вместо того, чтобы создавать проблемы, самопересекающаяся траектория, как показано на рисунке 37 в данном случае является полезной, позволяя транспортному средству эффективно разворачиваться или менять направление движения в ограниченном пространстве.

3.3.6 Ускорение расчетов за счет фиксированного горизонта при решении QP методом OSQP

В реализуемом алгоритме на каждом такте управления решается задача квадратичного программирования (QP), формируемая по результатам дискретизации модели движения и задания квадратичного функционала качества. Требование работы в режиме реального времени накладывает ограничения на допустимую вычислительную сложность решения, в связи с чем

отдельно рассматриваются приемы сокращения времени расчета.

Размерность задачи QP определяется длиной горизонта N , размерностями вектора состояния n_x и вектора управления n_u , а также шагом дискретизации dt . При включении в вектор оптимизируемых переменных z всех состояний и управлений на горизонте имеет место линейное масштабирование

$$\dim(z) \sim (N + 1)n_x + Nn_u, \quad (3.54)$$

при этом число ограничений также растет линейно по N . Если в процессе работы горизонтом управляют динамически (например, в зависимости от длины доступного сегмента траектории), то изменяются размерность z , число ограничений и, как следствие, структура разреженных матриц P и A в канонической постановке OSQP. Данное обстоятельство приводит к необходимости повторной инициализации задачи и выполнению подготовительных операций, оказывающих существенное влияние на суммарное время решения.

В разработанной реализации статические препятствия не вводятся в задачу QP в виде набора отдельных линейных неравенств, число которых росло бы с увеличением количества объектов. Вместо этого на стадии геометрической обработки сцены для каждой дискретной точки горизонта s_i формируется допустимый интервал поперечного смещения

$$[d_{\min,i}^{\text{obs}}, d_{\max,i}^{\text{obs}}], \quad (3.55)$$

который затем пересекается с дорожным интервалом

$$[d_{\min,i}^{\text{road}}, d_{\max,i}^{\text{road}}] \quad (3.56)$$

с учетом защитных зазоров согласно (3.142).

В результате ограничения свободного пространства в QP имеют вид

$$d_{\min,i} \leq d_i \leq d_{\max,i}, \quad i = 0, \dots, N, \quad (3.57)$$

и, следовательно, количество соответствующих скалярных ограничений остается равным $2(N + 1)$ и определяется только горизонтом N .

При изменении числа препятствий в цикле управления изменяются

преимущественно численные значения границ (l, u) (и, при необходимости, линейный член q), тогда как разреженная структура матриц P и A сохраняется неизменной, что позволяет переиспользовать факторизацию ККТ-матрицы и обеспечивает предсказуемое время решения при фиксированном N . С целью исключения указанных накладных расходов в постановке фиксируются параметры $\{N, n_x, n_u, dt\}$, что обеспечивает постоянную размерность QP. Практически это реализуется формированием на каждом такте ровно $N+1$ опорных точек (состояний) в плановом окне; при достижении конца пути последние точки опорной последовательности дублируются. В результате сохраняются неизменными размерность оптимизации и разреженная структура матриц задачи.

Для численного решения используется решатель OSQP, основанный на методе чередующихся направлений множителей (Alternating Direction Method of Multipliers, ADMM). В итерационном процессе OSQP основной вычислительной операцией является многократное решение линейной системы, соответствующей условиям оптимальности задачи (ККТ-структура). Данная система имеет блочную форму, формируемую по матрицам P и A (а также по параметрам регуляризации ADMM); для иллюстрации может быть записана в виде

$$K \begin{bmatrix} z \\ \lambda \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \end{bmatrix}, \quad K = \begin{bmatrix} P + \sigma I & A^\top \\ A & -\frac{1}{\rho} I \end{bmatrix}, \quad (3.58)$$

где λ — вектор множителей, σ и ρ — параметры метода. На практике OSQP применяет прямой разреженный метод решения данной системы с предварительной факторизацией (например, $LDL^\top$). Факторизация является наиболее затратной стадией, поскольку по вычислительной сложности существенно превосходит последующие операции обратного хода, выполняемые на каждой итерации.

Следовательно, ускорение достигается за счет обеспечения неизменности «тяжелой» части постановки. При неизменных матрицах P и A факторизация матрицы K может быть выполнена один раз и далее переиспользована на последующих тактах управления. Если же на каждом такте изменяются P или A (например, вследствие изменения структуры ограничений либо изменения размерности задачи), требуется повторная факторизация, что приводит к резкому росту времени решения. В связи с этим постановка задачи формируется

таким образом, чтобы в цикле управления обновлялись только линейный член q и границы ограничений l, u , тогда как матрицы P и A сохранялись постоянными вместе с их разреженной структурой. При таком подходе вычислительная стоимость одного такта сводится к обновлению нескольких векторов и выполнению итераций ADMM с уже подготовленной факторизацией.

Дополнительно в задачах MPC может использоваться теплый старт, при котором решение предыдущей QP-задачи подается в OSQP как начальное приближение для следующего такта. Однако в рамках настоящего подраздела количественно оценивается именно эффект от фиксированного горизонта и неизменности структуры матриц P и A , позволяющих переиспользовать факторизацию ККТ-матрицы.

Таким образом, ускорение расчетов при решении QP методом OSQP в рассматриваемой постановке обеспечивается за счет фиксированного горизонта оптимизации и связанной с ним неизменности размерности задачи и структуры матриц.

Основная сводка по решению заданной задачи оптимизации представлена в таблице 12.

Таблица 12 – Основная сводка по решению заданной задачи оптимизации

Число точек траектории	300	100
Время выполнения	25 мс	5 мс
Время выполнения (фиксированная длина траектории)	17 мс	2 мс

Полученный эффект подтверждается экспериментальными измерениями времени выполнения (таблица 12), где при использовании фиксированной длины траектории время расчета сокращается с 25 до 17 мс (на 32%) для случая 300 точек и с 5 до 2 мс (на 60%) для случая 100 точек.

3.4 Выводы

В Главе 3 разработан и обоснован алгоритмический подход к формированию локальной траектории движения беспилотного транспортного средства на основе методов выпуклой оптимизации, обеспечивающий получение траектории, пригодной для исполнения контуром управления при наличии геометрических ограничений дорожной инфраструктуры.

1. Сформулирована базовая постановка сглаживания опорной траектории в декартовой системе координат в виде задачи квадратичного программирования, в которой целевая функция задает компромисс между точностью следования опорной линии и требованиями к плавности за счет штрафов на кривизну и скорость ее изменения. Показано, что изменение весовых коэффициентов позволяет целенаправленно изменять характер траектории, однако отсутствие ограничений на свободное пространство ограничивает применимость постановки в задачах безопасного планирования.

2. Предложено расширение QR-постановки в декартовой системе координат путем введения коридорных ограничений на допустимую область движения. Установлено, что параметризация коридора в виде интервальных ограничений по координате при фиксированной (монотонной) оси обладает ограниченной областью применимости и не обеспечивает корректную обработку маневров со сложной геометрией, а также не учитывает габариты транспортного средства в явном виде.

3. Для устранения указанных ограничений разработана постановка планирования траектории в криволинейной системе координат Френе, в которой дорожные границы естественным образом задаются ограничениями на поперечное смещение относительно опорной линии, а переменные оптимизации имеют непосредственную геометрическую интерпретацию.

4. Реализован метод учета габаритов транспортного средства при формировании ограничений проходимости на основе аппроксимации кузова набором покрывающих окружностей и построения допустимого коридора в координатах Френе с последующим включением границ проходимости в виде линейных неравенств задачи квадратичного программирования.

5. Введены практические модификации постановки для учета статических

препятствий и повышения робастности планируемой траектории, в том числе раздельное задание защитных зазоров относительно границ дороги и относительно препятствий, что обеспечивает согласование требований безопасности с ограниченной шириной дорожного коридора. Для сценария резкого поворота показано, что модификация функционала с членом J_{turn} увеличивает минимальный зазор со стороны внутренней границы с $\approx 0,25$ до $\approx 0,80$ м.

6. Показано, что при наличии ограничений проходимости допускается формирование траекторий с самопересечением в декартовой системе координат, что может быть использовано для реализации маневров разворота и смены направления движения при сохранении допустимости траектории относительно геометрических ограничений.

7. Для обеспечения вычислительной реализуемости в режиме реального времени обосновано ускорение расчетов при решении QP методом OSQP за счет фиксированного горизонта оптимизации, обеспечивающего неизменность размерности задачи и структуры разреженных матриц с возможностью переиспользования факторизаций. Экспериментально показано сокращение времени расчета с 25 до 17 мс (на 32%) для случая 300 точек и с 5 до 2 мс (на 60%) для случая 100 точек.

Сущность результатов, полученных в главе 3, состоит в разработке метода синтеза локальной траектории движения автомобиля как наземного транспортно-технологического средства, реализуемого в составе функциональной подсистемы планирования автоматизированной системы управления движением. Формируемая локальная траектория является заданием для контура траекторного управления автомобилем и определяется с учетом текущего состояния транспортного средства, геометрии проезжей части, положения статических препятствий, габаритных параметров автомобиля, а также ограничений на кривизну траектории и скорость ее изменения, обусловленных возможностями рулевого управления и требованиями к динамической реализуемости движения. Научная новизна разработанного метода заключается в совокупности взаимосвязанных решений: параметрическом сглаживании опорной линии на основе совместной минимизации отклонения от нее, кривизны траектории и скорости изменения

кривизны; формировании по карте расстояний связного выпуклого полигона допустимого движения с явным учетом границ проезжей части, статических препятствий и габаритной аппроксимации автомобиля и последующем представлении его границ в виде системы линейных ограничений; синтезе итоговой траектории в системе координат Френе в форме выпуклой задачи квадратичного программирования с непосредственным учетом ограничений на поперечное положение автомобиля, кривизну траектории и скорость ее изменения. В отличие от подходов, в которых сглаживание опорной линии, учет свободного пространства и проверка габаритной проходимости выполняются раздельно, предложенный метод обеспечивает согласованное формирование допустимого коридора и последующую оптимизацию траектории в рамках единой алгоритмической схемы. Применение фиксированного горизонта планирования обеспечивает неизменность размерности и структуры разреженных матриц задачи, предсказуемость вычислительных затрат и возможность реализации метода на бортовом вычислителе автомобиля в режиме реального времени. Таким образом, полученные результаты представляют собой совершенствование функциональной системы управления движением колесного транспортного средства, направленное на обеспечение безопасности, геометрической и динамической реализуемости, а также плавности автоматического движения в условиях дорожных ограничений.

ГЛАВА 4. Экспериментальные исследования разработанных алгоритмов движения на физическом автомобиле

В настоящей главе приведены результаты экспериментальной верификации разработанных алгоритмов планирования и генерации траекторий, выполненной на беспилотном автомобиле Lada Vesta. Экспериментальная проверка организована в виде последовательности этапов, обеспечивающих поэтапное снижение рисков и повышение воспроизводимости результатов: предварительная верификация в симуляционной среде, проверка работоспособности в городской среде при реальном дорожном движении. Целью испытаний является подтверждение вычислительной реализуемости алгоритмов в режиме реального времени, а также оценка качества движения по критериям безопасности, плавности и точности исполнения.

4.1 Испытательная платформа и архитектура системы

Объектом испытаний является легковой автомобиль Lada Vesta (рисунок 38), оснащенный бортовым вычислительным комплексом, набором сенсоров и исполнительными интерфейсами, обеспечивающими автономное формирование управляющих воздействий. Функциональная архитектура реализована в среде ROS и включает контуры восприятия, локализации, планирования и управления; обмен с исполнительными механизмами выполняется через шину CAN. В целях безопасного проведения экспериментов предусматривается присутствие водителя-испытателя, готового выполнить переход к ручному управлению при возникновении нештатных ситуаций.



Рисунок 38 – Беспилотный автомобиль *Lada Vesta* с установленной системой автономного управления движением

Состав измерительно-вычислительного комплекса приведен в табл. 13.

Таблица 13 – Состав и параметры измерительно-вычислительного комплекса

Подсистема	Компонент	Параметры
Вычисления	Бортовой вычислитель	промышленный компьютер (Intel Xeon E3, 32 ГБ RAM, GPU NVIDIA RTX 4090)
Локализация	GNSS/RTK, IMU, одометрия	двухантенный RTK-GNSS приемник Trimble BX992 совместно с колесными энкодерами
Восприятие	Лидар	лидар Hesai Pandar64 (64 луча, 10 Гц, дальность 200 м)
Исполнители	Рулевое управление, тяга/тормоз	электропривод рулевого управления

4.2 Протокол испытаний и условия проведения

4.2.1 Требования к условиям проведения

Испытания выполнялись при метеорологических и дорожных условиях, не приводящих к деградации видимости и сцепных свойств покрытия. При необходимости допускается расширение диапазона условий.

Таблица 14 – Требования к условиям проведения испытаний

Параметр	Требование
Осадки	отсутствие либо слабые осадки при сохранении видимости
Видимость	не менее 1000 м
Температура воздуха	+5...+20 °
Освещенность	дневной свет;
Покрытие	сухое асфальтобетонное;

4.3 Виртуальная верификация в симуляторе Panosim

В ходе исследований в симуляционной среде Panosim была построена настраиваемая модель автономного транспортного средства и окружающей инфраструктуры. На этапе задания топологии дорожной сети в Panosim загружается карта местности, в которой задаются геометрия дорог и пересечений, а также параметры покрытия, разметки и светофорного регулирования.

Далее выполняется сборка виртуального автомобиля (рисунок 39): выбираются шасси, колесные пары, подвеска и приводы, а также конфигурируются датчики и исполнительные механизмы, что обеспечивает воспроизведение архитектуры тестируемой платформы. После этого задаются динамические характеристики автомобиля (рисунок 40): масса, моменты инерции, коэффициенты сцепления шин, параметры аэродинамического сопротивления и модели силовых агрегатов.

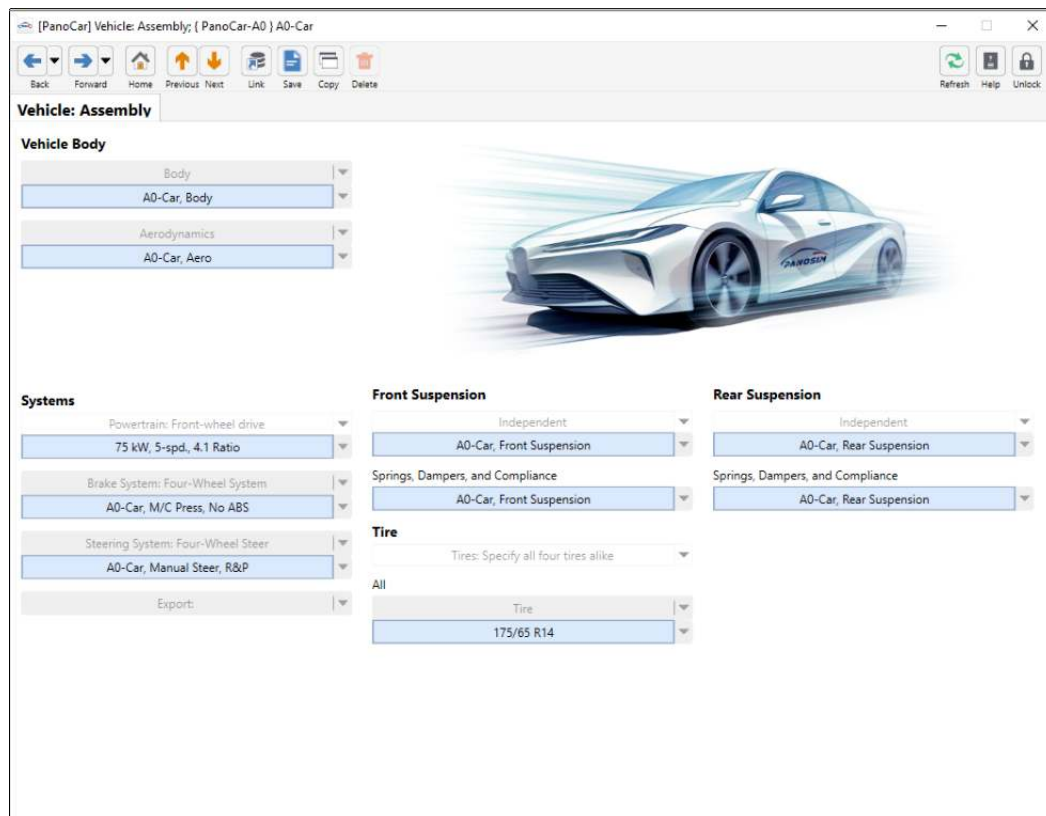


Рисунок 39 – Задание автомобиля в симуляторе Panosim

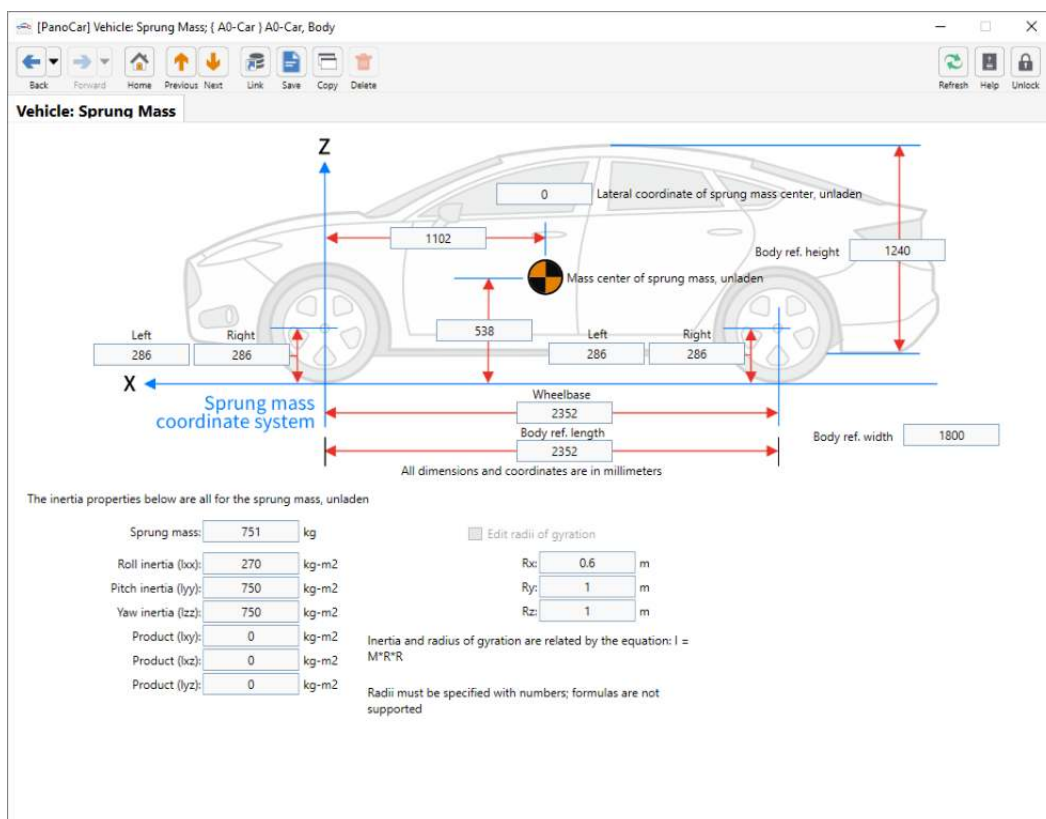


Рисунок 40 – Задание динамики автомобиля в симуляторе Panosim

Загружается карта тестового сценария (рис. 41).

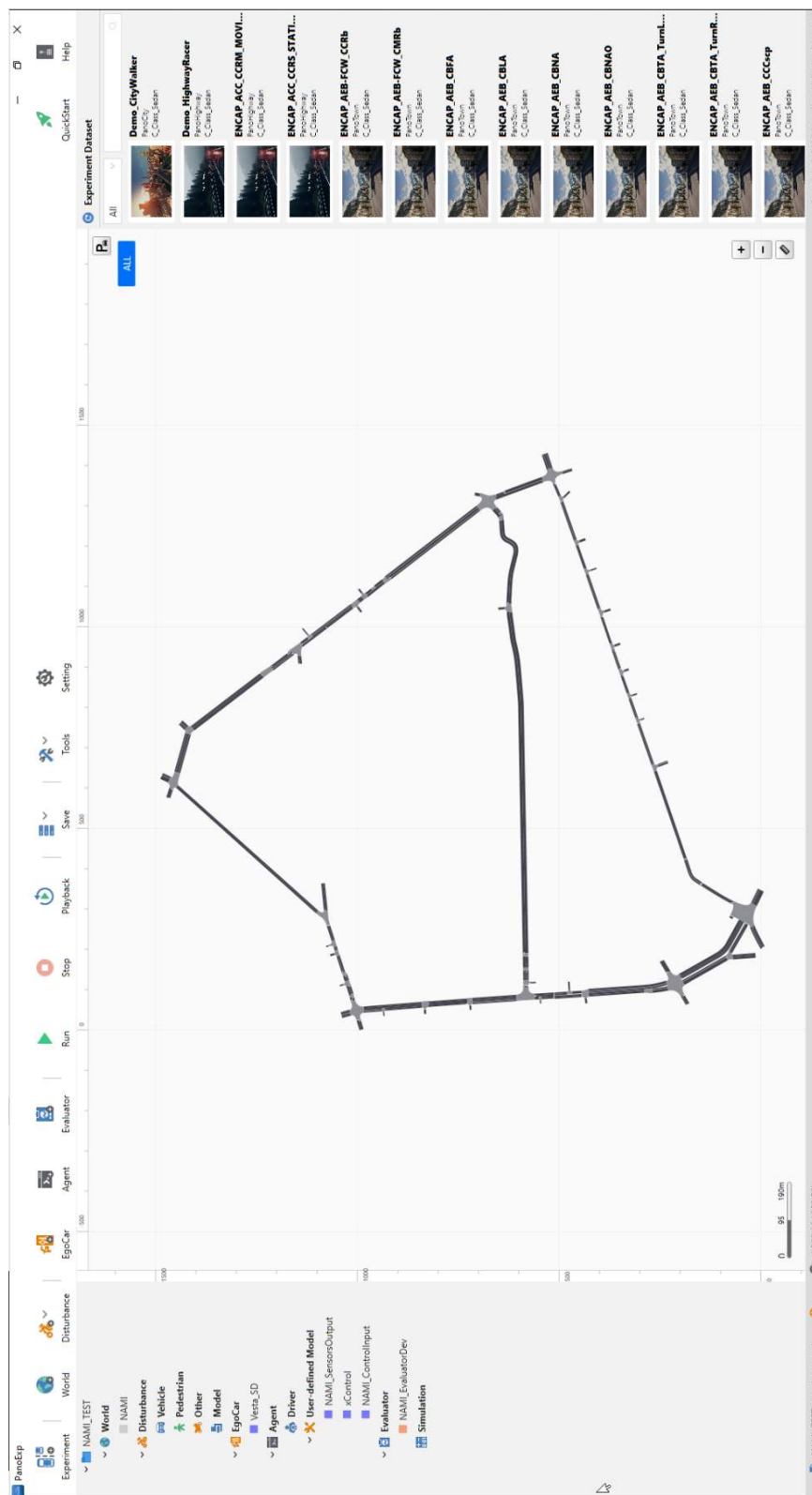


Рисунок 41 – Задание карты в симуляторе Panosim

Для визуальной оценки работы алгоритмов восприятия и планирования создается вид от «третьего лица», отображающий поля зрения сенсоров и элементы локальной карты окружения. На рис. 42 приведен пример траектории объезда статичного препятствия в полосе движения.

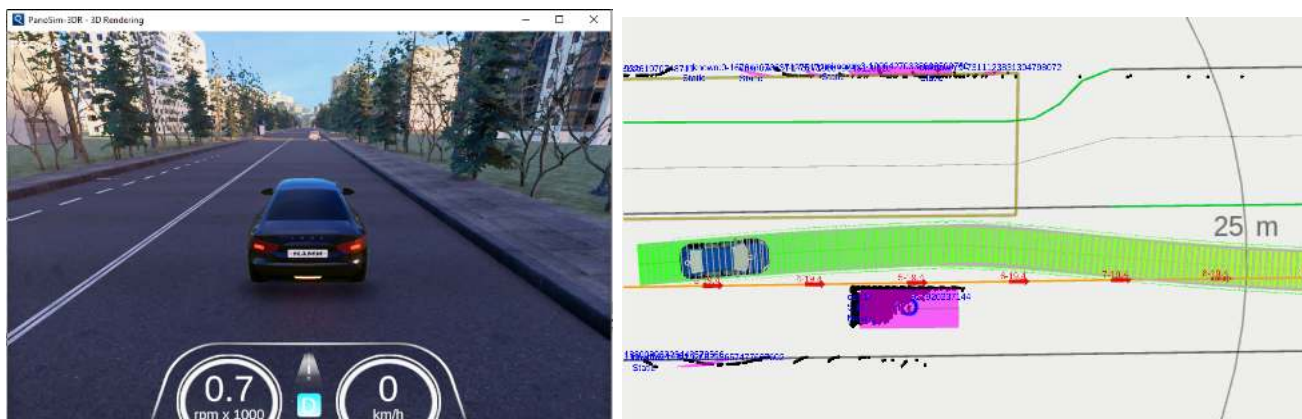


Рисунок 42 – Объезд статичного препятствия в полосе движения в симуляторе Panosim

В ходе проведенного эксперимента были получены оценочные численные характеристики сформированной траектории, представленные в таблице 15.

Таблица 15 – Оценочные характеристики траектории, полученные в ходе эксперимента

Показатель	Значение	Примечание
Скорость автомобиля v	43 км/ч (11,94 м/с)	в момент построения траектории
Максимальная кривизна $\kappa_{\max}$	$\approx 0,022 \text{ м}^{-1}$	соответствует маневру уклонения
Минимальный радиус кривизны $R_{\min}$	$\approx 45,5 \text{ м}$	$R_{\min} = 1/\kappa_{\max}$
Минимальное расстояние до границы свободного пространства $\Delta_{\min}$	$\approx 0,45 \text{ м}$	в наиболее стесненной зоне
Максимальное поперечное ускорение $a_{y,\max}$	$\approx 3,14 \text{ м/с}^2$	$a_y = v^2 \kappa$
Поперечное ускорение на прямом участке a_y	0–0,5 м/с ²	вне активного маневра

При скорости движения автомобиля 43 км/ч максимальная кривизна траектории составила приблизительно $\kappa_{\max} \approx 0,022 \text{ м}^{-1}$, что соответствует минимальному радиусу кривизны порядка 45,5 м. Минимальное расстояние до границы свободного пространства составило около 0,45 м, тогда как с противоположной стороны сохранялся запас порядка 1,3 м. Траектория была

построена в пределах допустимого коридора движения с сохранением необходимого зазора до его границ. Максимальное поперечное ускорение достигало приблизительно $3,14 \text{ м/с}^2$, что соответствует допустимому уровню динамической нагрузки и подтверждает физическую реализуемость полученной траектории. На участках, близких к прямолинейному движению, величина поперечного ускорения не превышала $0,5 \text{ м/с}^2$, что дополнительно подтверждает отсутствие избыточных колебаний и устойчивый характер движения.

Для формализации результатов этапа 4.3 проверка выполнялась по критериям геометрической допустимости, динамической реализуемости и качества траектории (таблица 16).

Таблица 16 – Результаты верификации алгоритма на симуляционном этапе

Проверяемое требование	Критерий	Результат	
Соблюдение геометрических ограничений коридора движения	$\Delta_{\min} > 0$ (нет пересечения границ допустимой области)	$\Delta_{\min} \approx 0,45 \text{ м}$	
Динамическая реализуемость на маневре объезда	$a_{y,\max} < 0,4g$ (прибл. $3,92 \text{ м/с}^2$)	$a_{y,\max} \approx 3,14 \text{ м/с}^2$	
Плавность геометрии траектории	Ограниченные значения кривизны и отсутствие резких скачков	$\kappa_{\max} \approx 0,022 \text{ м}^{-1}$, $R_{\min} \approx 45,5 \text{ м}$	

Таким образом, симуляционный этап используется как предварительная верификация корректности формирования ограничений, устойчивости решения задач оптимизации и соответствия траекторий базовым требованиям реализуемости до перехода к натурным испытаниям.

4.4 Проведение тестовых испытаний в городской среде

Алгоритм протестирован в городской среде на маршрутах, включающих различные типы дорожной инфраструктуры: регулируемые перекрестки, дороги с различным числом полос, участки с припаркованными автомобилями и пешеходные переходы. В ходе испытаний система работала в режиме автономного управления при присутствии водителя-испытателя.

Для тестирования выбран маршрут на дорогах общего пользования в непосредственной близости от НАМИ. На рис. 43 представлена визуализация маршрута на картографической подложке, на рис. 44 — маршрут, построенный по высокоточной карте, а на рис. 45 — пример визуализации работы системы в реальном времени.

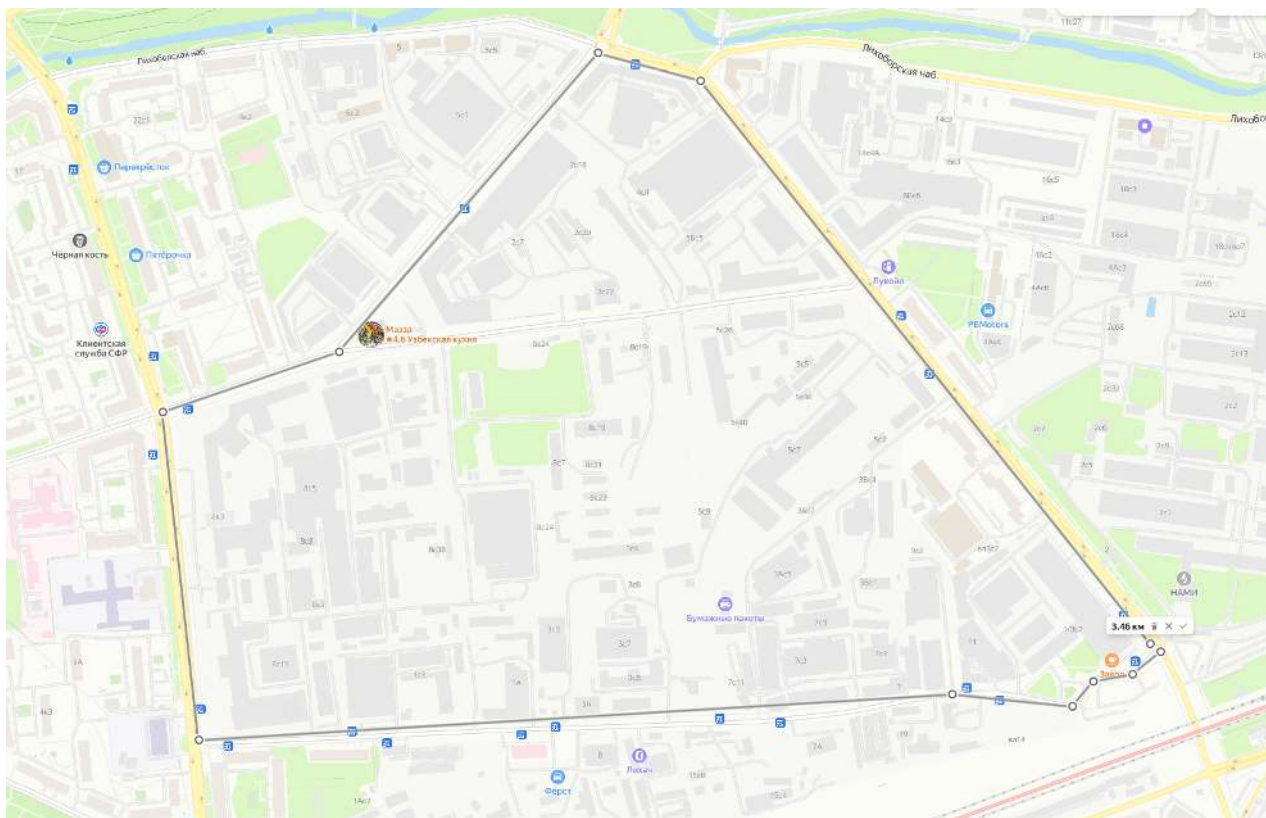


Рисунок 43 – Визуализация тестового маршрута на Яндекс Картах



Рисунок 44 – Визуализация маршрута, построенного с помощью высокоточных карт

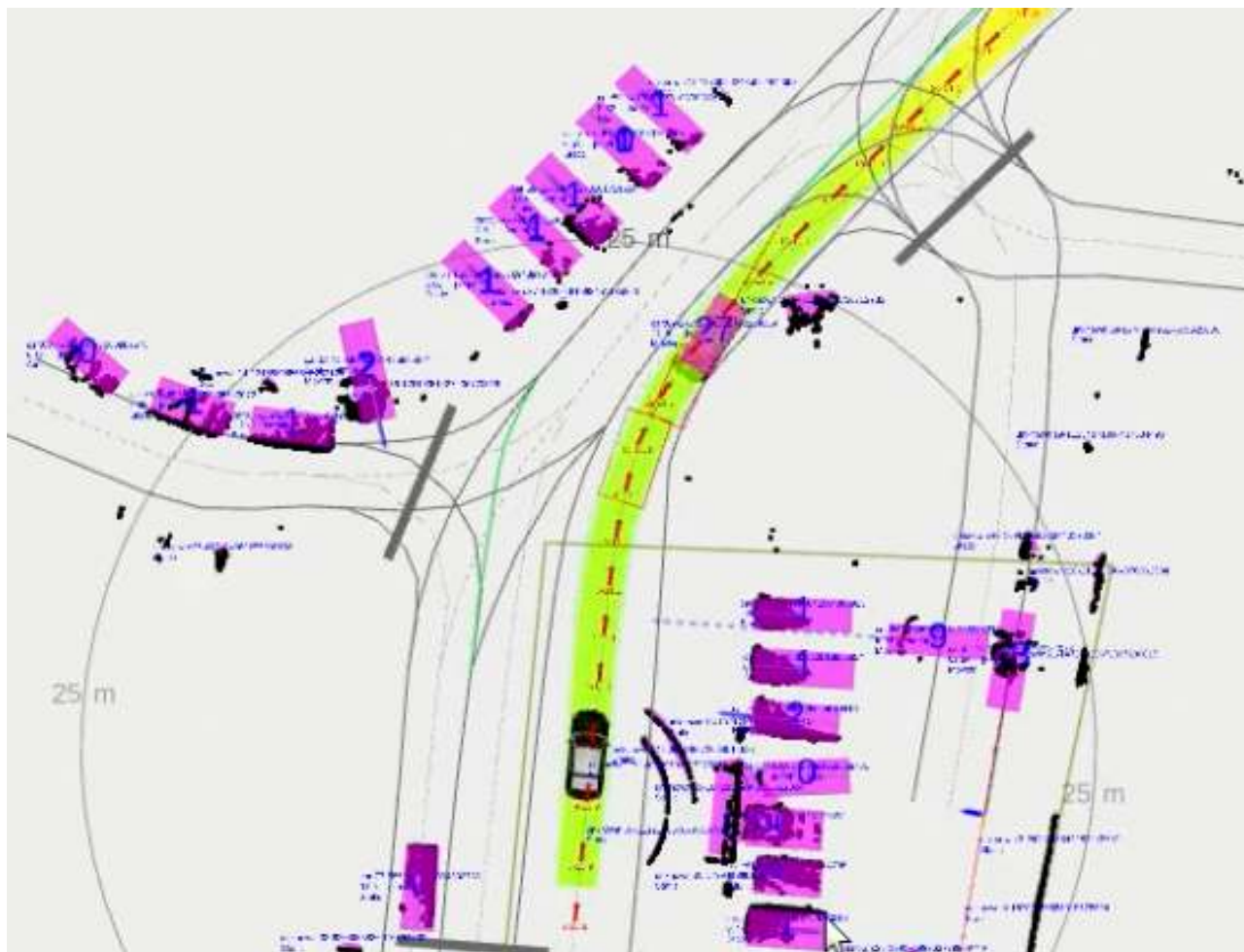


Рисунок 45 – Визуализация работы системы в реальном времени

На рис. 46 представлены телеметрические зависимости, полученные во время замкнутого городского пробега протяженностью около 3,46 км. В табл. 17 приведены ключевые характеристики плановой траектории, скоростного профиля и ускорений. Пилообразная форма кривой скорости соответствует городскому режиму. Пик продольного замедления порядка -6 м/с^2 соответствует эпизоду интенсивного торможения.

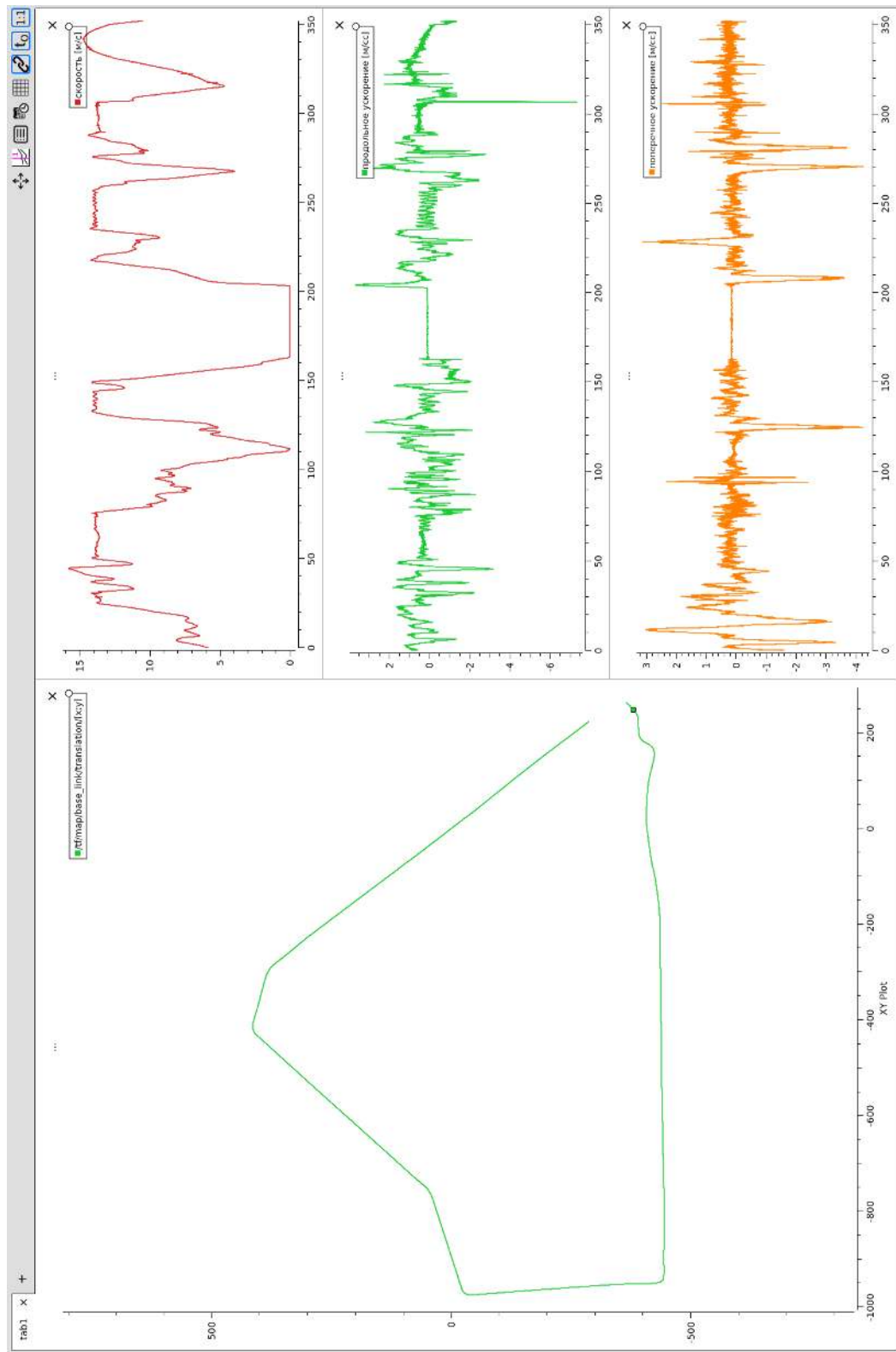


Рисунок 46 – Проезд маршрута, на котором выполнялось тестирование в городской среде

Таблица 17 – Сводные характеристики плановой траектории, скоростного режима и ускорений

Условное обозначение	Описание	Значение
Параметры плановой траектории		
SC	Система координат	ENU, м
x_{range}	Диапазон оси x	$[-1000; +250]$ м
y_{range}	Диапазон оси y	$[-600; +600]$ м
L_{loop}	Длина петли (периметр)	$\approx 3,46$ км
R_{min}	Минимальный радиус поворота	$\gtrsim 80$ м
Параметры скоростного режима		
v_{max}	Максимальная скорость	≈ 15 м/с (54 км/ч)
$\bar{v}$	Средняя скорость	$\approx 9,1$ м/с
$T_{v=0}$	Доля времени на остановках	$\approx 29\%$
$a_{\text{long}}^{\uparrow}$	Среднее ускорение при разгоне	$\approx 0,86$ м/с ²
Параметры продольного ускорения		
$a_{\text{long}}^{\text{min}}$	Минимальное продольное ускорение (торможение)	-6 м/с ²
$a_{\text{long}}^{\text{max}}$	Максимальное продольное ускорение (разгон)	$+3$ м/с ²
Параметры поперечного ускорения		
$a_{\text{lat}}^{\text{min}}$	Минимальное поперечное ускорение	$-3,8$ м/с ²
$a_{\text{lat}}^{\text{max}}$	Максимальное поперечное ускорение	$+3,2$ м/с ²
g_{comfort}	Порог комфортного поперечного ускорения по критерию Euro NCAP Comfort	$0,4 g (\approx 3,92 \text{ м/с}^2)$

Амплитуда a_{lat} не превышает $0,35 g$ (см. таблица 17), то есть удовлетворяет пороговому значению $0,4 g$, используемому в рамках оценок комфортности. Пики поперечного ускорения коррелируют с участками повышенной кривизны при сниженной скорости движения, что соответствует принципу согласования «высокая кривизна — малая скорость».

Скорость в заезде ограничивалась диапазоном до 50 км/ч; эпизод $v_{\text{max}} \approx 54$ км/ч носит кратковременный характер и не влияет на выводы, поскольку анализ качества выполнялся по статистикам и выделенным эпизодам движения. Пики ускорений, выходящие за типовой диапазон автономного режима, соответствуют интервалам вмешательства водителя-испытателя (перехват управления).

Для количественной верификации алгоритма построения безопасной траектории дополнительно были выделены эпизоды автономного движения, не содержащие перехвата управления со стороны водителя-испытателя. По указанным эпизодам рассчитывались показатели минимального расстояния до

границ допустимого коридора, минимального расстояния до статических препятствий, времени генерации траектории и поперечного ускорения. Результаты сведены в таблице 18.

Таблица 18 – Количественные показатели работы алгоритма локального планирования траектории по результатам городских испытаний

Показатель	Значение	Примечание
Минимальное расстояние до границы проезжей части	$\geq 0,3$ м	выдерживание допустимого дорожного коридора
Минимальное расстояние до статического препятствия в полосе движения	$\geq 0,5$ м	безопасный объезд препятствия
Среднее время генерации траектории	≈ 50 мс	вычислительная реализуемость в реальном времени
Максимальное по модулю поперечное ускорение	$\leq 2,5$ м/с ²	для выделенных эпизодов автономного объезда

Из данных таблицы 18 следует, что разработанный алгоритм локального планирования в ходе городских испытаний обеспечивал построение траектории, удовлетворяющей одновременно геометрическим, динамическим и вычислительным ограничениям. Во всех рассмотренных эпизодах минимальное расстояние до границ проезжей части сохранялось на уровне не менее 0,3 м, а до статических препятствий, расположенных в полосе движения, — не менее 0,5 м. Среднее время генерации траектории составило около 50 мс, что соответствует частоте порядка 20 Гц и подтверждает возможность применения разработанного алгоритма в контуре управления реального времени. При этом максимальное по модулю поперечное ускорение для выделенных эпизодов автономного объезда не превышало 2,5 м/с², что свидетельствует о динамической реализуемости формируемых траекторий и их соответствии требованиям плавности движения.

Далее приведен пример анализа траектории при выполнении маневра объезда «змейки». На рисунке 47 показана визуализация состояний, управлений и динамических параметров на интервале маневра.

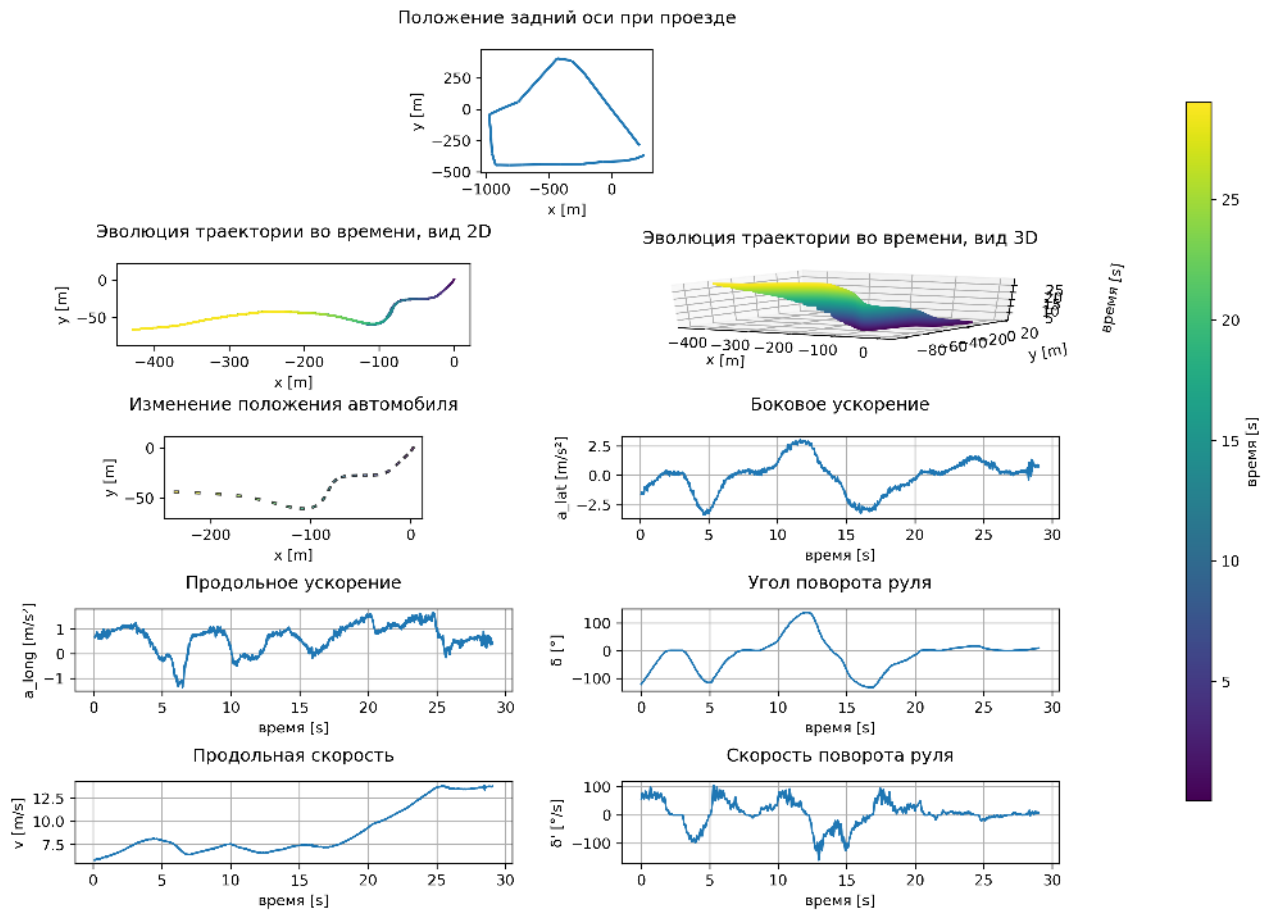


Рисунок 47 – Визуализация параметров движения при проезде «змейки» в городской среде

Представленные на рис. 47 зависимости иллюстрируют формирование непрерывных профилей управлений и ускорений без выраженных разрывов.

Для количественной оценки динамических и управляющих параметров, зарегистрированных при выполнении маневра «змейка», определены характеристики скорости движения, продольного и поперечного ускорений, а также параметры рулевого управления. Сводные результаты обработки логов эпизода представлены в таблице 19.

Таблица 19 – Сводные метрики маневра «змейка» по данным обработанных логов эпизода

Показатель	Обозначение	Оценка
Длительность эпизода	T	≈ 30 с
Скорость движения: среднее значение и диапазон изменения	$\bar{v}, v_{\min} \dots v_{\max}$	$\approx 10,0$ м/с; $8,5 \dots 11,5$ м/с
Диапазон изменения продольного ускорения	$a_{x,\min} \dots a_{x,\max}$	$-0,9 \dots +1,2$ м/с ²
Диапазон изменения поперечного ускорения	$a_{y,\min} \dots a_{y,\max}$	$-2,5 \dots +2,5$ м/с ²
Квантиль 0,95 модуля поперечного ускорения	$q_{0,95}(a_y)$	$\approx 2,2$ м/с ²
Диапазон изменения рулевого угла	$\delta_{\min} \dots \delta_{\max}$	$-110 \dots +105$ град
Максимальная по модулю скорость изменения рулевого угла	$\max \dot{\delta} $	≈ 100 град/с

Поперечные ускорения на интервале маневра находятся в диапазоне порядка $\pm 2,5$ м/с², а продольная скорость сохраняется вблизи выбранного уровня порядка 10 м/с, что соответствует постановке маневра объезда препятствий при ограничениях на комфорт и реализуемость управления.

На основании совокупности результатов городских испытаний выполнена проверка соответствия фактических показателей установленным целевым требованиям диссертационной работы. Результаты сопоставления представлены в таблице 20.

Таблица 20 – Подтверждение целевых требований диссертации по результатам городских испытаний

Требование	Контролируемый показатель	Фактический результат	Статус
Удержание траектории в геометрических границах проезжей части	Минимальное расстояние от габарита автомобиля до границы проезжей части не менее 0,3 м	$d_{\text{road}}^{\min} \geq 0,3$ м (табл. 18)	выполнено
Безопасный объезд статических препятствий	Минимальное расстояние от габарита автомобиля до препятствия не менее 0,5 м	$d_{\text{obs}}^{\min} \geq 0,5$ м (табл. 18)	выполнено
Вычислительная реализуемость в реальном времени	Среднее время формирования траектории не более 100 мс	$\bar{t}_{\text{gen}} \approx 50$ мс (около 20 Гц)	выполнено
Динамическая реализуемость траектории в автономных эпизодах	Максимальное по модулю поперечное ускорение не более 2,5 м/с ²	$ a_y _{\max} \leq 2,5$ м/с ² (табл. 18)	выполнено

4.4.1 Сравнение с альтернативными подходами и вклад разработанного алгоритма

Сравнение алгоритма выполнено по критериям внедрения на реальном автомобиле с использованием результатов обзорного анализа (глава 1) и экспериментальных данных глав 3–4 (см. таблица 21).

Таблица 21 – Сравнение разработанного метода с альтернативными подходами

Критерий	Граф./решет.	Sampling	NLP	Разработанный метод
Единая постановка с учетом границ полосы, габаритов ТС и статических препятствий	частично; обычно с постобработкой	частично; часто с постобработкой	да	да, в единой QP-постановке
Предсказуемость времени расчета	зависит от дискретизации и эвристик	вариативна, недетерминирована	вариативна (сходимость зависит от старта)	высокая за счет фиксированного горизонта и неизменной структуры матриц
Проверка на физическом автомобиле по метрикам безопасности и режима реального времени	как правило, требует отдельной адаптации	как правило, требует отдельной адаптации	ограничена вычислительной стоимостью	подтверждена: $\min dist_{road} \geq 0,3 \text{ м}$, $\min dist_{obs} \geq 0,5 \text{ м}$, $\bar{t}_{gen} \approx 50 \text{ мс}$

Для количественной интерпретации различий с опубликованными результатами выполнено сопоставление по времени расчета локальной траектории (таблица 22). В качестве значения для разработанного метода использован результат городских испытаний $\bar{t}_{gen} \approx 50 \text{ мс}$ (таблица 18).

Таблица 22 – Количественное сопоставление с опубликованными решениями

Источник	Сопоставляемый показатель	Значение в статье	Наш результат	Отличие
Li et al. (GPM) [74]	Время расчета траектории (17 узлов интерполяции)	0,392 с	0,050 с	быстрее на 87,2%
Li et al. (GPM) [74]	Время расчета траектории (31 узел интерполяции)	0,846 с	0,050 с	быстрее на 94,1%
Dolgov et al. (Hybrid A*) [75]	Полный цикл перепланирования (поиск + сглаживание + интерполяция)	0,050–0,300 с	0,050 с	на уровне нижней границы; до 83,3% быстрее по верхней границе

Таблица 22 показывает, что по метрике времени расчета траектории разработанный метод обеспечивает выраженный выигрыш относительно

опубликованного решения на основе гауссова псевдоспектрального подхода: от 87,2% до 94,1% в приведенных конфигурациях. Для ‘Hybrid A*’ сопоставление следует трактовать как ориентировочное: в работе Dolgov et al. приведено время полного цикла перепланирования, включающего поиск, сглаживание и интерполяцию, тогда как в данной работе используется метрика среднего времени генерации локальной траектории. Тем не менее, даже в такой постановке разработанный алгоритм находится на уровне нижней границы опубликованного диапазона ‘Hybrid A*’ и дает выигрыш до 83,3% относительно его верхней границы. Сравнение корректно только при сопоставимых сценариях: скорость движения, плотность препятствий и геометрия коридора должны быть близкими, а также должны совпадать горизонт и дискретизация планирования.

Совокупность результатов виртуальной верификации и городских испытаний показывает, что разработанный алгоритм обеспечивает построение безопасной траектории движения в реальном времени с учетом геометрии проезжей части, границ полос движения и статических препятствий. На этапе имитационного моделирования подтверждена корректность формирования ограничений и устойчивость оптимизационной постановки, а на этапе дорожных испытаний — сохранение минимальных безопасных зазоров до границ проезжей части и препятствий, вычислительная реализуемость с характерным временем генерации порядка 50 мс и отсутствие выхода по поперечному ускорению за пределы принятого комфортного диапазона для автономного режима объезда. Тем самым экспериментально подтверждена работоспособность разработанного алгоритма локального планирования в составе автоматизированной системы управления движением автомобиля.

4.5 Выводы

В главе 4 выполнена экспериментальная проверка разработанных алгоритмов на физической платформе с поэтапной схемой верификации: симуляция — городская среда. Симуляционный этап подтвердил возможность воспроизведения физико-информационных процессов и применимость алгоритмов для отладки сценариев с ограничениями, дорожными границами и статическими препятствиями. В городской среде показана работоспособность системы на маршруте на дорогах общего пользования с характерным режимом *stop&go*; приведены сводные характеристики плановой траектории и динамических параметров движения, а также пример эпизода маневра объезда с анализом профилей управлений и ускорений. Для автономного движения установлено, что минимальное расстояние до границ проезжей части составляло не менее 0,3 м, до статических препятствий — не менее 0,5 м, среднее время генерации траектории — около 50 мс, а максимальное по модулю поперечное ускорение не превышало $2,5 \text{ м/с}^2$. В совокупности с результатами главы 3 это позволяет зафиксировать и количественные улучшения, полученные в работе: в сценарии резкого поворота минимальный зазор со стороны внутренней границы увеличен с $\approx 0,25$ до $\approx 0,80$ м, а при решении QP методом OSQP за счет фиксированного горизонта время расчета сокращено с 25 до 17 мс (на 32%) для 300 точек и с 5 до 2 мс (на 60%) для 100 точек. При сопоставимых сценариях по времени расчета локальной траектории разработанный метод также превосходит опубликованный гауссов псевдоспектральный подход на 87,2–94,1% и находится на уровне нижней границы диапазона *Hybrid A** (50–300 мс), обеспечивая выигрыш до 83,3% относительно его верхней границы. Полученные результаты подтверждают вычислительную реализуемость алгоритма в режиме реального времени и его способность формировать безопасные и динамически реализуемые траектории движения автомобиля в городской среде, что непосредственно соответствует цели диссертационной работы.

ОСНОВНЫЕ РЕЗУЛЬТАТЫ И ВЫВОДЫ ПО РАБОТЕ

1. Проведены анализ и классификация методов локального планирования и управления траекторным движением высокоавтоматизированных и беспилотных автомобилей. Методы планирования систематизированы с выделением графовых и решеточных, выборочных, параметрических, оптимизационных и обучаемых подходов; методы управления классифицированы на геометрические, линейно-квадратичные и модельно-предиктивные. К основным ограничениям известных подходов отнесены зависимость результата от дискретизации и начального приближения, необходимость сглаживания, вариативность времени вычисления, косвенный учет кривизны и габаритов автомобиля, вычислительная сложность нелинейной оптимизации и затрудненная верификация обучаемых моделей. Геометрические и линейно-квадратичные регуляторы ограниченно учитывают характеристики рулевого привода и поперечной динамики. Сформулированы требования к разрабатываемым методам: требуется совместный учет геометрии дороги, габаритов автомобиля, статических препятствий, кинематических и динамических ограничений; гладкость и реализуемость траектории; предсказуемое время вычисления и работа в реальном времени. Обосновано применение координат Френе и выпуклой квадратичной оптимизации. На основании проведенной аналитической работы поставлена цель диссертационного исследования, на ее основании сформирован перечень основных задач исследования, обозначен объект исследования.

2. Обоснован выбор и адаптирован комплекс математических моделей движения автомобиля для локального планирования и траекторного управления. Пересчет состояния в глобальном пространстве выполнен по кинематической велосипедной модели в декартовых координатах. Для поперечного управления использована динамическая велосипедная модель, преобразованная к ошибкам следования, линеаризованная относительно опорной траектории и дискретизированная по времени. Для локального планирования получена пространственная кинематическая модель в координатах Френе с состоянием $z = [d, e_\theta, \kappa]^T$ и управлением κ' , определяющим изменение кривизны по дуговой координате. Опорная линия аппроксимирована кубическим В-сплайном для

устойчивого определения курса и кривизны. Сформированы ограничения по границам дороги, препятствиям и габаритам автомобиля; кривизне и скорости ее изменения; углу поворота колес и скорости его изменения; поперечному ускорению. Модели и ограничения приведены к форме квадратичного программирования.

3. Разработан метод траекторного управления на основе MPC с линеаризованной дискретной моделью поперечной динамики, описывающей поперечную и курсовую ошибки и скорости их изменения. Критерий качества (квадратичный функционал) учитывает ошибки следования, угол поворота колес и его приращение, обеспечивая баланс точности, интенсивности и плавности управления. Непосредственно заданы ограничения на угол и скорость поворота колес и поперечное ускорение. При горизонте 35 шагов и шаге 0,1 с среднее время вычисления составило 2,386–2,703 мс, 95-процентный квантиль — 4,590–5,053 мс, что подтверждает вычислительную реализуемость в реальном времени. Сравнение Pure Pursuit, LQR и MPC на S-образной траектории при скоростях 3, 5, 10 и 13,88 м/с показало отсутствие регулятора, минимизирующего ошибку во всех режимах: LQR обеспечил наименьшую RMSE при 3, 5 и 13,88 м/с, Pure Pursuit — при 10 м/с; для LQR и MPC при этой скорости RMSE составила 0,038 м. MPC принят в качестве базового метода, так как в разработанной постановке непосредственно минимизирует ошибки следования, величину и приращение управляющего воздействия при одновременном соблюдении ограничений рулевого привода и поперечной динамики. В реализованных вариантах Pure Pursuit и LQR такой совместный учет критерия качества и ограничений не обеспечивается.

4. Разработан метод и алгоритм синтеза локальной траектории на основе выпуклой квадратичной оптимизации. Сначала опорная траектория сглаживается в декартовых координатах путем минимизации отклонения, кривизны и скорости ее изменения. Затем по карте расстояний формируется связный выпуклый полигон допустимого движения, задаваемый линейными ограничениями с учетом габаритов автомобиля. На заключительном этапе в координатах Френе решается QP-задача с учетом границ коридора, габаритов, кривизны и скорости ее изменения. Функционал согласует точность следования, плавность и безопасные зазоры. Исследовано влияние весовых коэффициентов на отклонение, кривизну и расстояние до границ коридора. В сценарии резкого

поворота минимальный зазор со стороны внутренней границы увеличен приблизительно с 0,25 до 0,80 м при сохранении выполнимости ограничений. Фиксированный горизонт и неизменная структура матриц сократили время решения OSQP с 25 до 17 мс для 300 точек и с 5 до 2 мс для 100 точек, соответственно на 32 и 60 %, подтвердив применимость метода в реальном времени.

Основное отличие предлагаемого подхода от ранее выполненных исследований состоит в разработанном способе сопряжения геометрического и оптимизационного этапов локального планирования. На первом этапе по данным о границах проезжей части и статических препятствиях формируется связный допустимый коридор движения с учетом габаритов автомобиля. На втором этапе границы полученного коридора представляются в виде линейных ограничений выпуклой задачи квадратичного программирования, в которой синтезируется траектория с ограничениями на кривизну и скорость ее изменения. Применение модифицированного функционала качества и фиксированного горизонта планирования обеспечивает одновременное соблюдение требований геометрической безопасности, динамической реализуемости и вычислительной эффективности в режиме реального времени.

5. Разработанные методы и алгоритмы реализованы в программно-алгоритмическом комплексе автоматизированной системы управления движением автомобиля «Лада Веста». Комплекс формирует геометрические и динамические ограничения, решает QP-задачи на фиксированном горизонте, генерирует локальную траекторию и управляющее воздействие рулевого привода по данным о дорожном коридоре и препятствиях. Верификация включала вычислительные эксперименты, моделирование в PanoSim и натурные испытания в городской среде. Для MPC значение RMSE поперечной ошибки составило 0,006–0,116 м в зависимости от скорости. При имитации объезда на скорости 43 км/ч получены максимальная кривизна около $0,022 \text{ м}^{-1}$, минимальный радиус 45,5 м, минимальный зазор 0,45 м и максимальное поперечное ускорение $3,14 \text{ м/с}^2$, не превышающее установленного ограничения. В натурных испытаниях расстояние от габарита автомобиля до границ дороги составляло не менее 0,3 м, до препятствий — не менее 0,5 м; среднее время генерации траектории — около 50 мс, максимальное поперечное ускорение — не более $2,5 \text{ м/с}^2$, продольное ускорение при объезде

— приблизительно от $-0,9$ до $+1,2$ м/с². Результаты подтвердили точность управления, реализуемость траекторий, безопасность объезда статических препятствий и вычислительную применимость комплекса в реальном времени.

Таким образом, поставленная цель диссертационной работы достигнута за счет разработки и экспериментальной верификации методов синтеза локальной траектории и закона траекторного управления автомобилем на основе математического моделирования и выпуклой оптимизации. Разработанные методы обеспечивают формирование и выполнение гладкой, геометрически допустимой и динамически реализуемой траектории в режиме реального времени с учетом геометрии проезжей части, габаритов автомобиля, статических препятствий, ограничений рулевого привода и поперечной динамики. Решение каждой из пяти поставленных задач подтверждено результатами теоретических исследований, вычислительных экспериментов, имитационного моделирования и натурных испытаний автомобиля «Лада Веста».

Рекомендации по применению. Разработанные алгоритмы целесообразно использовать при наличии достоверной локализации, актуальной модели дорожной инфраструктуры и условий движения, соответствующих принятым допущениям о сцепных свойствах покрытия и видимости. При расширении диапазона эксплуатации необходимо фиксировать внешние условия и учитывать их при интерпретации метрик безопасности и плавности. Внедрение разработанных методов и программно-алгоритмического комплекса в другую аппаратную платформу должно сопровождаться повторной настройкой весовых коэффициентов функционала, проверкой ограничений рулевого привода и контрольными испытаниями на сценариях объезда статических препятствий в контролируемой и городской среде.

СПИСОК СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ

Сокращение	Расшифровка
АСУД	автоматическая система управления движением автомобиля
БТС	беспилотное транспортное средство
ТС	транспортное средство
СПВ	система помощи водителю
ПДД	правила дорожного движения
ADAS	Advanced Driver-Assistance Systems (продвинутые системы помощи водителю)
PP	Pure Pursuit (алгоритм геометрического преследования)
LQR	Linear Quadratic Regulator (линейно-квадратичный регулятор)
MPC	Model Predictive Control (модельно-предиктивное управление)
QP	Quadratic Programming (квадратичное программирование)
OSQP	Operator Splitting Quadratic Program (решатель задач QP)
ADMM	Alternating Direction Method of Multipliers (метод чередующихся направлений множителей)
RMSE	Root Mean Square Error (среднеквадратическая ошибка)

СПИСОК ЛИТЕРАТУРЫ

1. *Singh, S.* Critical Reasons for Crashes Investigated in the National Motor Vehicle Crash Causation Survey : тех. отч. / National Highway Traffic Safety Administration. — 2015. — DOT HS 812 115. — URL: <https://crashstats.nhtsa.dot.gov/Api/Public/Publication/812115> (дата обр. 01.10.2025).
2. Российская газета. Почти 90% ДТП в 2023 году произошло по вине водителей [Электронный ресурс] / Российская газета. — URL: <https://rg.ru/2024/06/21/pochti-90-dtp-v-2023-godu-proizoshlo-po-vine-voditelej.html> (дата обр. 01.10.2025).
3. Securing America's Future Energy (SAFE). America's Workforce and the Self-Driving Future: Society [Электронный ресурс] / Securing America's Future Energy (SAFE). — URL: <https://avworkforce.secureenergy.org/society/> (дата обр. 01.10.2025).
4. *Ветлинский, В., Осипов, А.* Автоматические системы управления движением автотранспорта. — Ленинград : ” Машиностроение,” Ленинградское отд-ние, 1986. — 120 с.
5. Перспективы развития беспилотного автотранспорта в дорожно-климатических условиях РФ / С. В. Бахмутов, С. Е. Бузников, А. М. Сайкин, Д. В. Ендачев // Безопасность колёсных транспортных средств в условиях эксплуатации. — 2019. — С. 505—514.
6. *Zhileykin, M., Kotiev, G., Nagatsev, M.* Comparative analysis of the operation efficiency of the continuous and relay control systems of a multi-axle wheeled vehicle suspension // IOP Conference Series: Materials Science and Engineering. T. 315. — IOP Publishing. 2018. — С. 012030.
7. *Якусевич, В. В., Нагайцев, М. В.* Перспективные задачи обеспечения подвижности беспилотных военных транспортных средств // Будущее машиностроения России. — 2020. — С. 18—21.

8. *Евграфов, В. В.* Динамика и управление движением колесных роботов : Дис. ... канд. физ.-мат. наук / Евграфов Владимир Владимирович. — Москва : Московский государственный университет им. М. В. Ломоносова, Механико-математический факультет, 2008.
9. *Куликов, И. А.* Совершенствование средств создания и исследования автомобилей с комбинированными энергоустановками с помощью технологий виртуально-физических испытаний : Дисс. ...канд. техн. наук / Куликов Илья Александрович. — Москва, 2016.
10. *Куликов, И. А., Ульченко, И., Чаплыгин, А.* Разработка и тестирование функций автоматического управления движением автомобиля // Технологии и компоненты наземных интеллектуальных транспортных систем. — 2019. — С. 441—444.
11. *Васин, П. А.* Совершенствование алгоритмов автоматического управления движением автомобиля посредством нейросетевых решений и анализа дорожной обстановки : Дисс. ...канд. техн. наук / Васин Павел Александрович. — Москва : ФГУП «НАМИ», 2022.
12. *Yakovlev, K., Andreychuk, A.* Towards time-optimal any-angle path planning with dynamic obstacles // Proceedings of the International Conference on Automated Planning and Scheduling. T. 31. — 2021. — С. 405—414.
13. *Андрейчук, А. А., Куликов, И. А.* Методы планирования траектории движения беспилотного автомобиля // Труды Международного автомобильного научного форума. — Москва, 2019. — С. 112—120.
14. *Ендачёв, Д. В.* Прогнозирование характеристик криволинейного движения беспилотного автомобиля : диссертация кандидата технических наук (05.05.03) / Ендачёв Денис Владимирович. — Москва : ФГУП «НАМИ», 2016.
15. *LaValle, S. M.* Planning Algorithms. — Cambridge : Cambridge University Press, 2006. — 842 с.
16. Tomlin, C. J. A Game Theoretic Approach to Controller Design for Hybrid Systems / C. J. Tomlin, J. Lygeros, S. Sastry // Proceedings of the IEEE. — 2000. — № 7. — С. 949—970. — DOI: <https://doi.org/10.1109/5.871303>.

17. Thrun, S. Stanley: The Robot That Won the DARPA Grand Challenge / S. Thrun, M. Montemerlo, H. Dahlkamp [и др.] // Journal of Field Robotics. — 2006. — № 9. — С. 661—692.
18. Leonard, J. J. A Perception-Driven Autonomous Urban Vehicle / J. J. Leonard, J. P. How, S. Teller [и др.] // Journal of Field Robotics. — 2008. — № 10. — С. 727—774. — DOI: <https://doi.org/10.1002/rob.20262>.
19. Saikin, A. Operating efficiency of active safety systems in the context of the Russian Federation / A. Saikin, S. Buznikov, A. A. Barashkov, A. A. Prokofiev // International Journal of Civil Engineering and Technology. — 2018. — № 10. — С. 1182—1191.
20. Беляков, Ф. Ю. Анализ методов построения траектории в зависимости от текущей дорожной обстановки / Ф. Ю. Беляков, В. В. Евграфов // Автомобильная промышленность. — 2025. — № 11. — С. 17—20. — DOI: <https://doi.org/10.36652/0005-2337/2025-11-17-20>. — EDN: AXCKBV.
21. Hart, P. E. A Formal Basis for the Heuristic Determination of Minimum Cost Paths / P. E. Hart, N. J. Nilsson, B. Raphael // IEEE Transactions on Systems Science and Cybernetics. — 1968. — № 2. — С. 100—107. — DOI: <https://doi.org/10.1109/TSSC.1968.300136>.
22. Stentz, A. The D* Algorithm for Real-Time Planning of Optimal Traverses : тех. отч. / Robotics Institute, Carnegie Mellon University. — Pittsburgh, PA, 10.1994. — CMU-RI-TR-94—37. — URL: <https://www.ri.cmu.edu/publications/the-d-algorithm-for-real-time-planning-of-optimal-traverses/> (дата обр. 03.06.2025).
23. Phillips, M., Likhachev, M. SIPP: Safe Interval Path Planning for Dynamic Environments // Proceedings of the IEEE International Conference on Robotics and Automation. — IEEE, 2011. — С. 5628—5635. — DOI: <https://doi.org/10.1109/ICRA.2011.5980306>.
24. Wang, S. State Lattice-based Motion Planning for Autonomous On-Road Driving : дис. ... канд. / Wang Shuiying. — Freie Universität Berlin, 2015.
25. LaValle, S. M. Rapidly-Exploring Random Trees: A New Tool for Path Planning : тех. отч. / Iowa State University. — Ames, IA, 1998. — TR 98—11.

26. Principles of Robot Motion: Theory, Algorithms, and Implementations / H. Choset, K. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. Kavraki, S. Thrun. — Cambridge, MA : MIT Press, 2005. — 603 c.
27. Karaman, S. Sampling-based Algorithms for Optimal Motion Planning / S. Karaman, E. Frazzoli // The International Journal of Robotics Research. — 2011. — № 7. — C. 846—894.
28. Paden, B. A survey of motion planning and control techniques for self-driving urban vehicles / B. Paden, M. Čáp, S. Z. Yong, D. Yershov, E. Frazzoli // IEEE Transactions on Intelligent Vehicles. — 2016. — № 1. — C. 33—55. — DOI: <https://doi.org/10.1109/TIV.2016.2578706>.
29. Optimal Trajectory Generation for Dynamic Street Scenarios in a Frenet Frame / M. Werling, J. Ziegler, S. Kammel, S. Thrun // Proceedings of the IEEE International Conference on Robotics and Automation (ICRA). — 2010. — C. 987—993.
30. Kelly, A. Reactive nonholonomic trajectory generation via parametric optimal control / A. Kelly, B. Nagy // The International Journal of Robotics Research. — 2003. — № 7—8. — C. 583—601. — DOI: <https://doi.org/10.1177/02783649030227008>.
31. Zhang, Y. Hybrid Trajectory Planning for Autonomous Driving in Highly Constrained Environments / Y. Zhang, H. Chen, S. L. Waslander, J. Gong, G. Xiong // IEEE Access. — 2018. — C. 32800—32819. — DOI: <https://doi.org/10.1109/ACCESS.2018.2845448>.
32. Angulo, B. Policy optimization to learn adaptive motion primitives in path planning with dynamic obstacles / B. Angulo, A. Panov, K. Yakovlev // IEEE Robotics and Automation Letters. — 2022. — № 2. — C. 824—831.
33. Jamal, M. FFStreams: Fast Search With Streams for Autonomous Maneuver Planning / M. Jamal, A. Panov // IEEE Robotics and Automation Letters. — 2024. — № 7. — C. 6752—6759.
34. Andreychuk, A. Multi-agent pathfinding with continuous time / A. Andreychuk, K. Yakovlev, P. Surynek, D. Atzmon, R. Stern // Artificial Intelligence. — 2022. — C. 103662.

35. *Yakovlev, K., Andreychuk, A.* Any-angle pathfinding for multiple agents based on SIPP algorithm // Proceedings of the International Conference on Automated Planning and Scheduling. T. 27. — 2017. — С. 586—594.
36. Improving continuous-time conflict based search / A. Andreychuk, K. Yakovlev, E. Boyarski, R. Stern // Proceedings of the AAAI Conference on Artificial Intelligence. T. 35. — 2021. — С. 11220—11227.
37. *Купреенко, А. И., Исаев, Х. М.-о., Михайличенко, С. М.* Автономные системы вождения в сельском хозяйстве // ИННОВАЦИИ И ТЕХНОЛОГИЧЕСКИЙ ПРОРЫВ В АПК. — 2020. — С. 249—254.
38. Dubins, L. E. On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents / L. E. Dubins // American Journal of mathematics. — 1957. — № 3. — С. 497—516.
39. *Scheuer, A., Fraichard, T.* Planning continuous-curvature paths for car-like robots // Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS'96. T. 3. — IEEE. 1996. — С. 1304—1311.
40. Pavlovskii, V. E. Mobile robots with two independent coaxial drive wheels: Dynamics and the schemes of control / V. E. Pavlovskii, V. Evgrafov, V. Pavlovskii, N. Petrovskaya // Moscow University Mechanics Bulletin. — 2009. — С. 8—13.
41. *Scheuer, A., Fraichard, T.* Continuous-curvature path planning for car-like vehicles // Proceedings of the 1997 IEEE/RSJ International Conference on Intelligent Robot and Systems. Innovative Robotics for Real-World Applications. IROS'97. T. 2. — IEEE. 1997. — С. 997—1003.
42. Fraichard, T. From Reeds and Shepp's to continuous-curvature paths / T. Fraichard, A. Scheuer // IEEE Transactions on Robotics. — 2004. — № 6. — С. 1025—1035.
43. *Bakolas, E., Tsiotras, P.* On the generation of nearly optimal, planar paths of bounded curvature and bounded curvature gradient // 2009 American Control Conference. — IEEE. 2009. — С. 385—390.
44. Piazzzi, A. n3-Splines for the Smooth Path Generation of Wheeled Mobile Robots / A. Piazzzi, C. G. L. Bianco, M. Romano // IEEE Transactions on Robotics. — 2007. — № 5. — С. 1089—1095.

45. *Boyd, S., Vandenberghe, L.* Convex Optimization. — Cambridge : Cambridge University Press, 2004. — 716 с.
46. Liu, C. The Convex Feasible Set Algorithm for Real Time Optimization in Motion Planning / C. Liu, C.-Y. Lin, M. Tomizuka // SIAM Journal on Control and Optimization. — 2018. — № 4. — С. 2712—2733. — DOI: <https://doi.org/10.1137/16M1091460>.
47. Owens, B. Как использовать диаграммы Вороного для управления ИИ [Электронный ресурс] / Owens, B. — URL: <https://habr.com/ru/articles/460813/> (дата обр. 08.10.2025).
48. Dolgov, D. Practical search techniques in path planning for autonomous driving / D. Dolgov, S. Thrun, M. Montemerlo, J. Diebel // ann arbor. — 2008. — № 48105. — С. 18—80.
49. Elfes, A. Using Occupancy Grids for Mobile Robot Perception and Navigation / A. Elfes // Computer. — 1989. — № 6. — С. 46—57.
50. *Thrun, S., Burgard, W., Fox, D.* Probabilistic Robotics. — Cambridge, MA : MIT Press, 2005. — 647 с.
51. Katrakazas, C. Real-time motion planning methods for autonomous on-road driving: State-of-the-art and future research directions / C. Katrakazas, M. Qudus, W.-H. Chen, L. Deka // Transportation Research Part C: Emerging Technologies. — 2015. — С. 416—442.
52. Wiyatno, R. R. Robotik - Ep.2: Intro to Motion Planning and Rapidly-exploring Random Tree (RRT) [Электронный ресурс] / Wiyatno, R. R. — URL: <https://rrwiyatn.github.io/blog/robotik/2020/05/16/rrt.html> (дата обр. 14.01.2025).
53. Евграфов, В. В. Основные принципы применения комбинированных цифровых карт для высокоавтоматизированных транспортных средств / В. В. Евграфов, К. Л. Кондратьев, Д. И. Редин, Д. В. Ендачев, Ф. Ю. Беляков // Московский транспорт. Наука и проектирование. — 2025. — № 4. — С. 9—17. — EDN: PZIYTH.
54. *Coulter, R. C.* Implementation of the Pure Pursuit Path Tracking Algorithm : тех. отч. / Carnegie Mellon University. — 1992.

55. *Anderson, B. D. O., Moore, J. B.* Optimal Control: Linear Quadratic Methods. — Englewood Cliffs, NJ : Prentice Hall, 1990. — 380 c.
56. Stellato, B. OSQP: An operator splitting solver for quadratic programs / B. Stellato, G. Banjac, P. Goulart, A. Bemporad, S. Boyd // Mathematical Programming Computation. — 2020. — № 4. — С. 637—672.
57. *Schultz, D. G., Melsa, J. L.* State functions and linear control systems. — New York : McGraw-Hill, 1967. — 435 c.
58. *Luenberger, D. G.* Introduction to Dynamic Systems: Theory, Models, and Applications. — New York : John Wiley & Sons, 1979. — 446 c.
59. *Skelton, R. E.* Dynamic systems control: linear systems analysis and synthesis. — New York : John Wiley & Sons, Inc., 1988. — 504 c.
60. *Camacho, E. F., Bordons, C.* Constrained model predictive control. — London : Springer, 2007. — 405 c.
61. *Qin, S. J., Badgwell, T. A.* An overview of industrial model predictive control technology // AIChE symposium series. T. 93. — New York, NY: American Institute of Chemical Engineers, 1971-c2002. 1997. — С. 232—256.
62. *Khalil, H. K.* Nonlinear systems. — 3-е изд. — Upper Saddle River, NJ : Prentice Hall, 2002. — 750 c.
63. Grewal, M. S. Appendix A: Software / M. S. Grewal, A. P. Andrews // Kalman Filtering: Theory and Practice with MATLAB. — 2015. — С. 593—603.
64. Kinematic and dynamic vehicle models for autonomous driving control design / J. Kong, M. Pfeiffer, G. Schildbach, F. Borrelli // 2015 IEEE Intelligent Vehicles Symposium, IV 2015, Seoul, South Korea, June 28 - July 1, 2015. — IEEE, 2015. — С. 1094—1099. — DOI: <https://doi.org/10.1109/IVS.2015.7225830>.
65. *Rajamani, R.* Vehicle Dynamics and Control. — 2-е изд. — New York, NY : Springer, 2012. — 516 c. — (Mechanical Engineering Series). — DOI: <https://doi.org/10.1007/978-1-4614-1433-9>.
66. *Gillespie, T. D.* Fundamentals of Vehicle Dynamics. — Warrendale, PA : Society of Automotive Engineers (SAE), 1992. — 495 c.

67. Huang, J. A Fusion Estimation Method for Tire-Road Friction Coefficient Based on Weather and Road Images / J. Huang, X. Chen, Q. Jin, P. Li // *Lubricants*. — 2025. — № 10. — С. 459. — DOI: <https://doi.org/10.3390/lubricants13100459>.
68. Беляков, Ф. Ю. Алгоритм сглаживания траектории движения беспилотного транспортного средства на основе квадратичной оптимизации / Ф. Ю. Беляков, В. В. Евграфов // *Труды НАМИ*. — 2025. — № 3(302). — С. 54—60. — EDN: AIPWTJ.
69. Беляков, Ф. Ю. Алгоритм сглаживания траектории движения беспилотного транспортного средства на основе квадратичной оптимизации с учётом ограничений на границы дороги / Ф. Ю. Беляков, В. В. Евграфов // *Известия МГТУ “МАМИ”*. — 2025. — № 3. — С. 156—163. — DOI: <https://doi.org/10.17816/2074-0530-689704>. — EDN: HWJOKW.
70. Trajectory planning under vehicle dimension constraints using sequential linear programming / M. G. Plessen, P. F. Lima, J. Mårtensson, A. Bemporad, B. Wahlberg // 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC). — IEEE. 2017. — С. 1—6.
71. *ISO. Road vehicles — Test track for a severe lane-change manoeuvre — Part 2: Obstacle avoidance : Standard* / International Organization for Standardization. — 2011.
72. *Pacejka, H. Tire and vehicle dynamics*. — Amsterdam : Elsevier, 2005. — 672 с.
73. Kam, H. RViz: a toolkit for real domain data visualization / H. Kam, S.-H. Lee, T. Park, C.-H. Kim // *Telecommunication Systems*. — 2015. — С. 1—9. — DOI: <https://doi.org/10.1007/s11235-015-0034-5>.
74. Li, Z. Obstacle Avoidance Trajectory Planning for Autonomous Vehicles on Urban Roads Based on Gaussian Pseudo-Spectral Method / Z. Li, X. Wu, W. Zhang, W. Yu // *World Electric Vehicle Journal*. — 2023. — № 1. — С. 7. — DOI: <https://doi.org/10.3390/wevj15010007>.
75. Dolgov, D. Path Planning for Autonomous Vehicles in Unknown Semi-structured Environments / D. Dolgov, S. Thrun, M. Montemerlo, J. Diebel // *The International Journal of Robotics Research*. — 2010. — № 5. — С. 485—501. — DOI: <https://doi.org/10.1177/0278364909359210>.

ПРИЛОЖЕНИЕ А. ПРОГРАММНАЯ РЕАЛИЗАЦИЯ АЛГОРИТМА ПОСТРОЕНИЯ ОГРАНИЧЕНИЙ ДЛЯ QR

```

1  #include <vector>
2  #include <cmath>
3  #include <algorithm>
4  namespace {
5
6  // ----- Константы поиска -----
7
8  // Радиус, внутри которого мы считаем, что окружность сталкивается с
   препятствием.
9  constexpr double kCollisionRadius = 0.5;
10
11 // Крупный шаг грубого поиска вдоль нормали m().
12 constexpr double kCoarseStep = 0.3;
13
14 // Длина тонкого шага уточняющего поиска m().
15 constexpr double kFineStep = 0.05;
16
17 // Сколько грубых шагов максимум мы делаем (6 м / 0.3 м = 20 шагов).
18 constexpr std::size_t kMaxCoarseSteps = static_cast<std::size_t>(6.0 /
   kCoarseStep);
19
20 } // namespace
21
22 // -----
23
24 std::vector<double> getClearance(const State& state, const Map& map)
25 {
26     // ----- Предварительные вычисления
   -----
27     // Нормаль к траектории: влево (left_angle) и вправо (right_angle).

```

```

28  const double left_angle = constrainAngle(state.heading + M_PI_2);
29  const double right_angle = constrainAngle(state.heading - M_PI_2);
30
31  // Позиция точки, с которой начинаем поиск.
32  const grid_map::Position start_pos(state.x, state.y);
33
34  // Если начальная точка уже слишком близко к препятствию —
    траектория заблокирована.
35  if (map.getObstacleDistance(start_pos) <= kCollisionRadius) {
36      return {0.0, 0.0};
37  }
38
39  // ----- Вспомогательная лямбда
    -----
40  // Возвращает >0, если точка на расстоянии step по нормали свободна;
41  // 0 или отрицательное значение означает столкновение.
42  auto distanceToFreeSpace = [&](double step, double angle) -> double {
43      const double x = state.x + step * std::cos(angle);
44      const double y = state.y + step * std::sin(angle);
45      return map.getObstacleDistance({x, y}) - kCollisionRadius;
46  };
47
48  // ----- Грубый поиск границы слева
    -----
49  double left_offset = 0.0; // положительное смещение метры()
50  for (std::size_t i = 0; i < kMaxCoarseSteps; ++i) {
51      left_offset += kCoarseStep;
52      if (distanceToFreeSpace(left_offset, left_angle) <= 0.0) {
53          left_offset -= kCoarseStep; // последний безопасный шаг
54          break;
55      }
56  }
57

```

```

58 // ----- Грубый поиск границы справа
   -----
59 double right_offset = 0.0; // положительное, позже сделаем
   отрицательным
60 for (std::size_t i = 0; i < kMaxCoarseSteps; ++i) {
61     right_offset += kCoarseStep;
62     if (distanceToFreeSpace(right_offset, right_angle) <= 0.0) {
63         right_offset -= kCoarseStep;
64         break;
65     }
66 }
67 right_offset = -right_offset; // справа смещения считаем
   отрицательными
68
69 // ----- Точный поиск тонкие( шаги)
   -----
70 auto refineBoundary = [&](double& offset, double angle, int sign) {
71     // sign = +1 для левой границы, -1 для правой тк(.. offset уже
   отрицателен)
72     const int fine_steps = static_cast<int>(kCoarseStep / kFineStep);
73     for (int i = 1; i < fine_steps; ++i) {
74         offset += sign * kFineStep;
75         if (distanceToFreeSpace(offset, angle) <= 0.0) {
76             offset -= sign * kFineStep; // возврат на последнюю безопасную
   точку
77             break;
78         }
79     }
80 };
81
82 refineBoundary(left_offset, left_angle, +1);
83 refineBoundary(right_offset, right_angle, -1);
84

```

```

85 // ----- Коррекция на фактический радиус кузова
   -----
86 // Окружность покрывающего круга меньше половины ширины
   автомобиля —
87 // нужно учесть эту разницу, чтобы границы относились к крайним
   точкам кузова.
88 const double body_radius_diff = FLAGS_car_width * 0.5 -
   kCollisionRadius;
89 left_offset -= body_radius_diff; // левая граница смещается наружу
   влево()
90 right_offset += body_radius_diff; // правая — наружу вправо(, но offset
   отрицателен)
91
92 // Если коридор схлопнулся — возвращаем нули, чтобы вызывающий
   код остановил путь.
93 if (left_offset < right_offset) return {0.0, 0.0};
94
95 // ----- Применяем защитный зазор
   -----
96 const double free_space = left_offset - right_offset; // ширина
   свободного коридора
97 constexpr double kMinSpace = 0.2; // минимально
   допустимая ширина
98 const double max_margin = std::max(0.0, (free_space - kMinSpace) /
   2.0);
99 const double safety_margin = std::min(FLAGS_safety_margin,
   max_margin);
100
101 left_offset -= safety_margin;
102 right_offset += safety_margin;
103
104 // ----- Результат
   -----
105 return {left_offset, right_offset};

```

```

106 }
107
108 // Смещения покрывающих окружностей метры(; задано во флагах)
109 const double d1;
110 const double d2;
111 const double d3;
112 const double d4;
113
114 void ReferencePathImpl::updateBoundsImproved() {
115     // 0. Быстрый выход, если опорная траектория пуста
116     if (reference_states_.empty()) {
117         return;
118     }
119
120     bounds_.clear();
121
122     // Постоянный массив смещений окружностей относительно задней
123     // оси
124     static const std::array<double,4> kD{d1, d2, d3, d4};
125
126     bool blocked = false; // флаг окончания цикла при тупике
127
128     for (const auto &rs : reference_states_) {
129         // === 1. Строим центры окружностей и их проекции на опорную
130         // линию ===
131         std::array<State,4> centre{}; // реальные центры c0..c3
132         std::array<State,4> proj{}; // проекции c00..c33
133         std::array<std::array<double,2>,4> clr{}; // {ub, lb} для каждой
134         // окружн.
135
136         for (size_t i = 0; i < 4; ++i) {
137             // 1.1 Реальный центр окружности
138             centre[i] = { rs.x + kD[i] * cos(rs.z),
139                         rs.y + kD[i] * sin(rs.z),

```

```

137             rs.z,
138             rs.s + kD[i] };
139
140         // 1.2 Проекция на опорную линию тот( же s, но y = 0)
141         proj[i] = getApproxState(centre[i].s);
142
143         // 1.3 Боковые зазоры относительно проекции
144         clr[i] = map_ ->getClearance(proj[i]);
145
146         // 1.4 Перенос зазоров к реальному центру
147         const double dy = global2Local(centre[i], proj[i]).y;
148         clr[i][0] += dy;
149         clr[i][1] += dy;
150
151         // 1.5 Проверка тупика: левый == правый
152         if (isEqual(clr[i][0], clr[i][1])) {
153             blocked = true;
154             break;
155         }
156     }
157     if(blocked) {
158         break;
159     }
160
161     // === 2. Сохраняем границы для QP ===
162     bounds_.emplace_back(CoveringCircleBounds{clr[0], clr[1], clr[2], clr
163     [3]});
164
165     // 3. reference_states_ не должен быть длиннее bounds_
166     reference_states_.resize(bounds_.size());
167 }

```

ПРИЛОЖЕНИЕ Б. АКТЫ О ВНЕДРЕНИИ

ОБЩЕСТВО С ОГРАНИЧЕННОЙ ОТВЕТСТВЕННОСТЬЮ «ИННОВАЦИОННЫЙ ЦЕНТР «КАМАЗ»

«Утверждаю»
Исполнительный директор
ООО «Инновационный центр КАМАЗ»
А. В. Горбатовский
« 16 » 04 _____ 2026 г.

АКТ

внедрения результатов диссертационной работы

Белякова Федора Юрьевича

«Алгоритм генерации локальной траектории движения автомобиля с учетом дорожных ограничений по маршруту движения» выполненной на соискание
учёной степени

кандидата технических наук по специальности 2.5.11 – Наземные
транспортно-технологические средства и комплексы

Настоящий акт подтверждает, что результаты диссертационного исследования соискателя ученой степени кандидата технических наук Белякова Федора Юрьевича на тему «Алгоритм генерации локальной траектории движения автомобиля с учетом дорожных ограничений по маршруту движения» были использованы ООО «Инновационный центр КАМАЗ» для построения траекторий при цифровом моделировании пространственного движения перспективных колёсных машин, а также отладке алгоритмов управления систем активной безопасности.

Руководитель службы
инженерных расчетов и
моделирования, д.т.н.

« » _____ **М.М. Жилейкин**
_____ 2026 г.

Руководитель службы
электрифицированных
автомобилей, к.т.н.

« » _____ **А.В. Климов**
_____ 2026 г.

- НАМИ -

ГОСУДАРСТВЕННЫЙ НАУЧНЫЙ ЦЕНТР РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ УНИТАРНОЕ ПРЕДПРИЯТИЕ

«Центральный ордена Трудового Красного Знамени Научно-исследовательский

автомобильный и автотранспортный институт «НАМИ» (ФГУП «НАМИ»)

ИНН/КПП 7711000924/774301001, ОКПО 00234703, ОГРН 1027739228406

ул. Автотранспортная, д. 2, г. Москва, 125438 • info@nami.ru; nami.ru

УТВЕРЖДАЮ
Зам. генерального директора
по науке ФГУП «НАМИ»,
д.т.н., профессор
Г.Т. Надарейшвили
«15» 04 2026 г.

АКТ ВНЕДРЕНИЯ

Результатов диссертационной работы Беякова Федора Юрьевича на тему:
«АЛГОРИТМ ГЕНЕРАЦИИ ЛОКАЛЬНОЙ ТРАЕКТОРИИ ДВИЖЕНИЯ
АВТОМОБИЛЯ С УЧЕТОМ ДОРОЖНЫХ ОГРАНИЧЕНИЙ ПО
МАРШРУТУ ДВИЖЕНИЯ»,

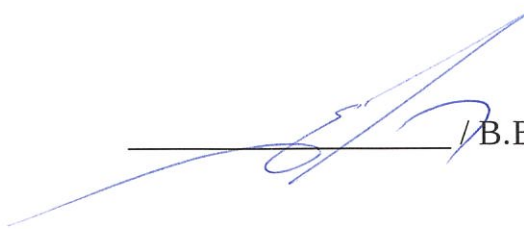
выполненной на соискание учёной степени кандидата технических наук по
специальности 2.5.11 – Наземные транспортно-технологические средства и
комплексы

Комиссия в составе председателя – директора центра «Интеллектуальные системы» к.ф.-м.н. В.В. Евграфова и членов комиссии – заместителя директора центра «Интеллектуальные системы» Н.П. Мезенцева и начальника отдела «Системы автоматического управления» управления «Автономные транспортные средства» центра «Интеллектуальные системы» Н.В. Мамонтова составила настоящий акт о том, что полученные в диссертационной работе Ф.Ю. Беякова результаты теоретических и экспериментальных исследований, а именно:

1. Программное обеспечение, реализующее автоматическое управление траекторией движения автомобиля с учётом окружающей дорожной обстановки;
2. Программное обеспечение, реализующее автоматическое следование по заданной траектории движения;

используются во ФГУП «НАМИ» в центре «Интеллектуальные системы» при исследованиях и разработке систем автоматического управления движением автомобилей.

Председатель комиссии


_____/ В.В. Евграфов
/

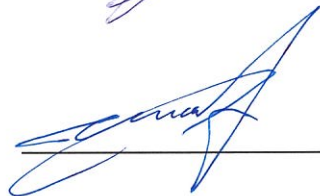
к.ф.-м.н.

Члены комиссии:

Заместитель директора центра


_____/ Н.П. Мезенцев
/

Начальник отдела


_____/ Н.В. Мамонтов
/