

Федеральное государственное автономное образовательное учреждение
высшего образования «Южно-Уральский государственный университет
(национальный исследовательский университет)»

На правах рукописи



Лисов Андрей Анатольевич

**РАЗРАБОТКА И ИССЛЕДОВАНИЕ СИСТЕМЫ ЭЛЕКТРОННОГО
ДИФФЕРЕНЦИАЛА ДЛЯ ЭЛЕКТРОАВТОМОБИЛЯ**

Специальность 2.4.2 «Электротехнические комплексы и системы»

Диссертация на соискание учёной степени
кандидата технических наук

Научный руководитель –
доктор технических наук,
профессор **Возмилов А.Г.**

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	4
ГЛАВА 1. СОСТОЯНИЕ ВОПРОСА. ЗАДАЧИ ИССЛЕДОВАНИЯ	11
1.1. Мировые тенденции развития в автомобильной отрасли	11
1.2. Состояние и перспективы развития автомобильной отрасли в России....	15
1.3. Анализ существующих вариантов компоновки тяговых электродвигателей электромобиля	19
1.4. Анализ эффективности существующих систем управления электродвигателем при поворотах и манёврах.....	22
1.4.1. Линейная система управления электроприводом	24
1.4.2. Нелинейная система управления электроприводом	25
Выводы по главе 1	29
ГЛАВА 2. ТЕОРЕТИЧЕСКИЕ АСПЕКТЫ РАЗРАБОТКИ СИСТЕМЫ ЭЛЕКТРОННОГО ДИФФЕРЕНЦИАЛА ДЛЯ УПРАВЛЕНИЯ ЭЛЕКТРОАВТОМОБИЛЕМ НА ОСНОВЕ НЕЙРОННОЙ СЕТИ	30
2.1. Математические модели линейной и нелинейной систем управления	30
2.2. Влияние инерционных сил на характер движения транспортного средства при поворотах и манёврах	33
2.3. Влияние погодных условий на эффективность управления транспортным средством.....	36
2.4. Математическая модель системы электронного дифференциала на основе НС	38
Выводы по главе 2.....	44
ГЛАВА 3. РАЗРАБОТКА ФИЗИЧЕСКОЙ МОДЕЛИ ЭЛЕКТРОМОБИЛЯ И МЕТОДИКИ ПОЛУЧЕНИЯ ЭКСПЕРИМЕНТАЛЬНЫХ ДАННЫХ НЕОБХОДИМЫХ ДЛЯ ОБУЧЕНИЯ НС	45
3.1. Разработка физической модели электромобиля	47
3.2. Методика получения экспериментальных данных для обучения НС	55
3.2.1. Методика получения экспериментальных данных	55
3.2.2. Методика обработки данных для создания датасета	62
3.2.3. Выбор типа нейронной сети для реализации системы электронного дифференциала.....	67
3.2.4. Разработка программного обеспечения для обучения нейронной сети на основе полученных экспериментальных данных	69
Выводы по главе 3.....	73
ГЛАВА 4. ПОЛУЧЕНИЕ ЭКСПЕРИМЕНТАЛЬНЫХ ДАННЫХ ДЛЯ ОБУЧЕНИЯ НС И НАСТРОЙКА ЕЁ АРХИТЕКТУРЫ	74

4.1. Получение экспериментальных данных для обучения НС с помощью физической модели электромобиля	74
4.2. Настройка архитектуры нейронной сети под полученные обработанные экспериментальные данные	77
4.2.1. Исследование влияния batch size на качество обучения нейронной сети	78
4.2.2. Исследование методов градиентного спуска для выбора метода оптимизации нейросети	84
4.2.3. Исследование влияния функции потерь на точность прогнозирования НС	86
4.3. Результаты обучения конечной архитектуры НС	88
4.4. Перспективы дальнейших исследований	94
Выводы по главе 4	99
ОБЩИЕ ВЫВОДЫ И РЕЗУЛЬТАТЫ	101
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	103
Приложение А. Принципиальная схема разработанной физической модели электромобиля и использованные компоненты.	118
Приложение Б. Печатная плата разработанной модели электромобиля с электронной дифференциальной системой	121
Приложение В. Внешний вид и размеры передаточного механизма для тягового привода модели электроавтомобиля	123
Приложение Г. ПО для обучения нейронной сети на подготовленных данных. .	124
Приложение Д. Дорожная карта диссертационной работы	129
Приложение Е. Свидетельство о регистрации программы для ЭВМ	130
Приложение Ж. Акт внедрения в производственный процесс	131

ВВЕДЕНИЕ

Актуальность темы и степень ее разработанности. В настоящее время во всём мире наблюдается активный переход от использования транспорта, оснащённого двигателями внутреннего сгорания (ДВС), к электротранспорту. В Российской Федерации данное направление подкреплено нормативно-правовыми актами, такими как распоряжение №2290-р от 23 августа 2021 года, касающееся концепции развития производства и использования электрического автомобильного транспорта до 2030 года, а также распоряжением от 28 декабря 2022 года №4261-р, утверждающим Стратегию развития автомобильной промышленности до 2035 года по производству электроавтомобилей.

Как и в автомобилях с ДВС, в электромобилях тяговый двигатель выполняет ключевую функцию, обеспечивая движение транспортного средства. В настоящее время, ведущие производители электроавтомобилей в качестве электропривода используют 2 и более тяговых электродвигателей. Для управления скоростью вращения электродвигателей используется специализированный алгоритм – система электронного дифференциала.

В настоящее время существуют линейные и нелинейные варианты реализации системы электронного дифференциала (СЭД), предложенные следующими авторами: Lee J.S, Yildirim M., Aggarwal A., Котина Д.А., Мосина М.Е., Бутенко В.Н., которые обладают рядом недостатков. Недостатком линейной системы электронного дифференциала является отсутствие учёта ряда факторов при поворотах электроавтомобиля (конструктивных особенностей транспортного средства и текущей скорости движения). Нелинейная система электронного дифференциала показывает более высокую эффективность управления по сравнению с линейной системой, поскольку учитывает не только угол поворота как линейная система, но и скорость электроавтомобиля, а также его конструктивные особенности, однако при повышенных скоростях движения выше 50 км/ч эффективность управления ухудшается, т.к. не учитываются возрастающее влияние на качество управления, инерционных сил, погодных условий и стиля вождения.

В связи с этим в данной работе предлагается использование идеи, высказанной Lee J.S. в 2000 году, касающейся применения искусственных нейронных сетей для решения вышеописанных проблем. Настоящая работа посвящена разработке системы управления тяговыми электродвигателями электроавтомобиля электронного дифференциала на основе нейронных сетей. В рамках данного исследования была разработана физическая модель электроавтомобиля для получения экспериментальных данных необходимых для обучения нейронной сети (НС), предложен метод обучения НС на основе разработанной модели электроавтомобиля и карт идеальных поворотов, а также осуществлён выбор и проектирование её архитектуры.

Целью диссертационной работы является разработка системы электронного дифференциала для управления электроавтомобилем при поворотах и манёврах на основе искусственных нейронных сетей.

Задачи исследования:

1. На основе анализа научно-технической литературы определить рациональную компоновку тягового электропривода электроавтомобиля.
2. Провести анализ эффективности существующих систем электронного дифференциала для управления тяговым электроприводом электроавтомобиля при движении на поворотах и манёврах.
3. Обосновать выбор нейронной сети для управления системой электронного дифференциала электроавтомобиля.
4. Разработать физическую модель электрического автомобиля для получения экспериментальных данных, необходимых для обучения нейронной сети.
5. Разработать методику получения экспериментальных данных на основе разработанной физической модели электроавтомобиля для обучения нейронной сети.
6. Выбрать архитектуру и обучить нейронную сеть для разработанной системы электронного дифференциала на основе полученных экспериментальных данных.

Объектом исследования являются процессы протекающие в системе управления тяговым электроприводом электроавтомобиля на основе системы электронного дифференциала при поворотах и манёврах.

Предметом исследования является система электронного дифференциала, основанная на искусственных нейронных сетях, для управления скоростью тяговых электродвигателей электроавтомобиля.

Научная новизна и практическая значимость состоит в том, что:

1. Разработан способ управления электроприводом электроавтомобиля при поворотах и манёврах, основанный на системе электронного дифференциала и нейронных сетях, позволяющий повысить эффективность управления транспортом.

2. Разработана методика получения экспериментальных данных для обучения искусственной нейронной сети, для управления электроприводом электроавтомобиля при поворотах и манёврах.

3. Использование системы электронного дифференциала на основе нейронных сетей для электроавтомобилей и беспилотных наземных транспортных средств позволяет повысить безопасность транспорта на дорогах.

4. Разработанную систему электронного дифференциала (СЭД) для электропривода электроавтомобиля возможно адаптировать к различным дорожным условиям и стилю вождения в реальном времени, обеспечивая эффективное управление скоростью автомобиля.

Методология и методы исследования.

В диссертационной работе использовались аналитические методы и сведения из теории электрических машин, преобразовательной техники и теории автоматического управления, методы экспериментального исследования. Были применены алгоритмы машинного обучения, правила разработки принципиальных схем и печатных плат использованы сведения из цифровой техники (программирование микроконтроллеров, цифровая фильтрация и передача данных).

Основные положения и результаты, выносимые на защиту

1. Результаты анализа эффективности существующих систем управления тяговым электроприводом электроавтомобиля при поворотах и манёврах.

2. Физическая модель электроавтомобиля для получения экспериментальных данных необходимых для обучения нейронной сети.

3. Методика получения экспериментальных данных необходимых для обучения искусственной нейронной сети на основе идеальных кривых поворота и разработанной физической модели электроавтомобиля.

4. Способ управления электроприводом электроавтомобиля при поворотах и манёврах, основанный на системе электронного дифференциала и нейронных сетях, для варианта компоновки с двумя электродвигателями сзади.

5. Результаты сравнения эффективности существующих способов управления скоростью электропривода электроавтомобиля с разработанной системой электронного дифференциала на основе нейронной сети.

Степень достоверности результатов работы

Степень достоверности разработанной системы электронного дифференциала на основе искусственных нейронных сетей для электроавтомобилей подтверждается результатами проведенных исследований и экспериментов. Использование методов анализа данных, включая статистические и алгоритмические подходы, обеспечивает надежность и точность полученных результатов. Проведенные эксперименты и тесты подтверждают функциональность и применимость системы в реальных условиях. Таким образом, на основании проведенных исследований и экспериментов можно утверждать, что разработанная система электронного дифференциала на основе нейронных сетей готова к практическому применению в современных электроавтомобилях.

Публикации и апробация работы.

В полном объёме работа докладывалась и обсуждалась на заседаниях кафедры: «Электропривод, мехатроника и электромеханика» ФГАОУ ВО «Южно-Уральский государственный университет (Национальный Исследовательский Университет)», г. Челябинск. Основные результаты работы докладывались и обсуждались на научно-технических конференциях, в том числе:

1. 2023. International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM). Международная научно-техническая конференция "Пром-Инжиниринг", 15-19 мая, г. Сочи.

2. 2024. Энерго- и ресурсосбережение в теплоэнергетике и социальной сфере. Международная научно-техническая конференция студентов, аспирантов, ученых, 5 апреля, г. Челябинск.

3. 2024. International Conference on Electrical Power Engineering (UralCon). Международная научно-техническая конференция «Электротехнические комплексы и системы», 27-29 сентября, г. Магнитогорск.

По материалам диссертации опубликовано 10 научных работ, из которых 6 в изданиях, входящих в перечень ВАК РФ, Scopus – 2, а также 1 свидетельство на программу для ЭВМ.

Личный вклад автора. Личный вклад автора состоит в постановке задач, разработанной методике, проведении экспериментов и обобщении результатов исследований. Все научные положения разработаны автором лично. Автор данной работы внес существенный вклад в разработку системы электронного дифференциала на основе искусственных нейронных сетей для электроавтомобилей. Он провел анализ существующих подходов и методов, выявил ключевые проблемы и решение задачи управления электроавтомобилем при поворотах и манёврах. Автор лично разрабатывал алгоритмы, программное обеспечение и проводил эксперименты.

Реализация и внедрение результатов:

1. В 2022 г. Получен грант от Фонда содействия развитию малых форм предприятий в научно-технической сфере» (Фонда содействия инновациям), «УМНИК» (рег. № 18080ГУ/2022) по тематике: «Разработка систем компьютерного зрения для распознавания дефектов на поверхности горячего листопроката».

2. Разработана и изготовлена экспериментальная модель электрического автомобиля с двумя независимыми двигателями сзади, с помощью которой

получены результаты экспериментальной проверки предложенных идей, подтверждающие теоретические положения.

3. Результаты диссертационной работы используются и внедрены в производственной деятельности ООО «ДСТ-Урал».

Структура и объем диссертации.

Диссертация состоит из введения, четырех глав, заключения, списка литературы из 143 наименований и 6 приложений. Основная часть изложена на 102 страницах, содержит 57 рисунков и 9 таблиц.

Глава 1. Состояние вопроса и задачи исследования. В данной главе представлен обзор современных тенденций в автомобильной отрасли, с акцентом на развитие электромобилей и беспилотных наземных транспортных средств. Анализируется компоновка тяговой системы электромобилей, включая основные компоненты и их взаимодействие. Представлены теоретические основы линейной и нелинейной системы управления, каждая из которых имеет свои ограничения и особенности применения в реальных условиях эксплуатации.

Глава 2. Теоретические аспекты разработки системы электронного дифференциала для управления электроавтомобилем на основе нейронной сети. Глава посвящена теоретическим аспектам разработки системы электронного дифференциала для управления электроавтомобилем на основе нейронной сети. Представлены математические модели линейной и нелинейной систем управления электроприводом, реализованные в среде Matlab/Simulink. Приведены математические выражения на основе модели семи степеней свободы для учёта влияния на автомобиль инерционных и аэродинамических сил при движении. Также рассмотрено влияние погодных условий на сцепление шин с дорогой и устойчивость автомобиля. Разработана структурная схема управления электроавтомобилем системой электронного дифференциала, основанной на НС, учитывающая дополнительные факторы, влияющие на управление электроприводом.

Глава 3. Разработка физической модели электромобиля и методики получения экспериментальных данных необходимых для обучения НС. В

данной главе описывается процесс разработки, проектирования и изготовления физической модели электромобиля для получения экспериментальных данных, включающее: разработку функциональной и принципиальной схемы тягового электропривода электромобиля, выбор компонентной базы для разработки печатной платы, а также разработку управляющего ПО для модели. В главе приведена методико-программная часть, содержащая в себе: разработку методики способа получения экспериментальных данных, анализ рекомендаций для создания набора обучающих данных, выбор типа наиболее подходящей нейронной сети для реализации СЭД и разработку программного обеспечения для обучения НС.

Глава 4. Получение экспериментальных данных для обучения НС и настройка её архитектуры. В данной главе представлены результаты экспериментов по реализации модели электромобиля и тестирования разработанной системы электронного дифференциала как при помощи вычислительных тестов, так и на практике. Исследуется влияние параметров и гиперпараметров на точность прогнозирования нейронной сети, разработанной для системы электронного дифференциала. В результате проведённых экспериментов был установлен оптимальный размер выборки (batch size), выбрана функция потерь и метод градиентного спуска для рациональной архитектуры НС. Рассмотрены результаты обучения нейронной сети, её эффективность в управлении системой электронного дифференциала для электроавтомобиля. А также рассмотрены перспективы дальнейших исследований и использования разработанной системы электронного дифференциала на основе НС в автомобильной отрасли.

Соответствие научной специальности: исследование, выполненное в рамках диссертационной работы, полностью соответствует паспорту специальности 2.4.2, в частности:

- первое научное положение соответствует п. 1;
- второе положение соответствуют п. 2;
- второе положение соответствуют п. 3;
- вся работа в целом, соответствует положению п.4.

ГЛАВА 1. СОСТОЯНИЕ ВОПРОСА. ЗАДАЧИ ИССЛЕДОВАНИЯ

1.1. Мировые тенденции развития в автомобильной отрасли

Анализ мировых тенденций развития в автомобильной отрасли показывает, что данное развитие идёт в нескольких направлениях:

1. **Электрификация.** Процесс электрификации заключается в том, что все механические системы (ДВС, коробка передач, дифференциал др.) заменяются на электротехнические системы (электродвигатель, электронный дифференциал, электронные системы управления) [43, 54, 89]. Кроме этого, бак с топливом заменяется на аккумуляторную батарею.

2. **Автономное вождение.** Компании вкладываются в разработку систем автономного управления, стремясь создать автомобили с минимальным участием водителя. Это включает технологии искусственного интеллекта, сенсоры и камеры, которые позволяют автомобилям самостоятельно анализировать дорожную обстановку [47, 111].

3. **Цифровизация.** Развитие умных технологий и интернета вещей (IoT) позволяет автомобилям становиться «умными», подключенными к сети транспортными средствами, что улучшает удобство вождения, безопасность и управление данными [86, 131].

4. **Экологическая устойчивость.** Стремление к снижению углеродного следа приводит к активному внедрению альтернативных видов топлива (например, водород), уменьшению выбросов CO₂ и переходу на экологически чистые производственные процессы [62].

Электромобили по сравнению с автомобилями с двигателем внутреннего сгорания (ДВС) обладают рядом существенных преимуществ и это прежде всего экологичность, широкий спектр возможностей использовать цифровые технологии, более высокий КПД (так при прохождении 100 км автомобилем пути в зависимости от вида использования первичной энергии КПД равно «зеленой энергетике» – 71,16, водорода – 29,67, энергии угля – 28,43, нефти – 21,44, см. рисунок 1.1),

повышенная безопасность за счёт уменьшения количества аварий (ДТП), обусловленных человеческим фактором и др.

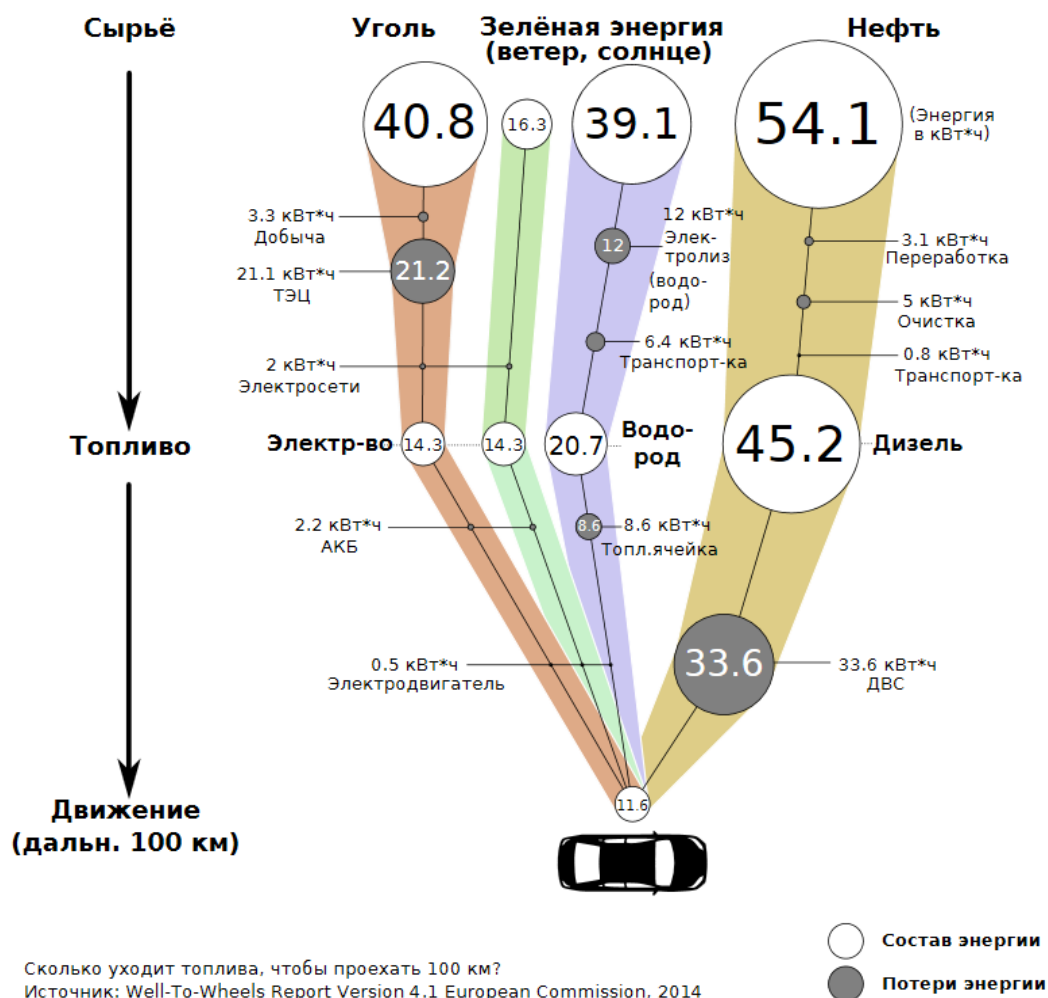


Рисунок 1.1 – Сравнение эффективности использования различных видов энергии для питания электромобиля

Как видно из рисунка 1.1 наибольший КПД – 71,16% электромобиль (ЭМ) имеет при зарядке АКБ от «зеленой энергетики», наименьший КПД – 21,44 % при первичном энергоносителе – нефть. Таким образом КПД электроавтомобилей в три раза выше, чем при использовании нефти [56].

Массовый выпуск электроавтомобилей впервые начался в 2016 году в США. На рисунке 1.2 показана динамика продаж электромобилей в США на период с 2016 по 2024 и прогноз от Statista до 2028 года. Динамика продаж электромобилей в мире в период с 2011 по 2021 год представлена на рисунке 1.3 (данные Statista).



Рисунок 1.2 – Статистика и прогноз продаж электромобилей в США в период с 2016 по 2028 годы

Анализ диаграммы рисунка 1.3 показывает, что наибольший объём продаж приходится на Китай, США и Германию – около 2/3 от всего объёма продаж.

Электромобили из-за своей способности к более гибкому управлению тяговым приводом позволяют разработать на своей базе устройство, обладающее элементами искусственного интеллекта и создать полностью или частично беспилотное транспортное средство.

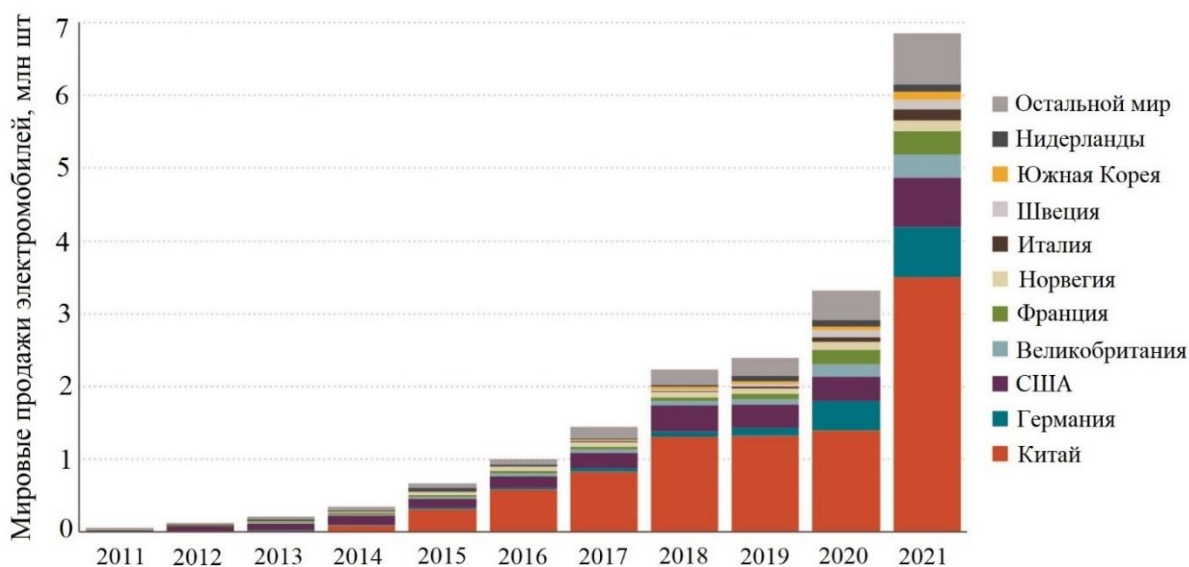


Рисунок 1.3 – Статистика продаж электромобилей странами мира в период с 2011 по 2021 год

Из рисунка 1.4 можно увидеть тенденцию развития в этом направлении, на нем приведены данные по количеству продаж беспилотных (частично) автомобилей в мире в период с 2022 года по прогнозный 2030 год. Количество беспилотных автомобилей, в мире на 2023 год составило около 21 тыс. штук, по прогнозам в ближайшие годы достигнет 125 тыс. штук (см. рисунок 1.4).

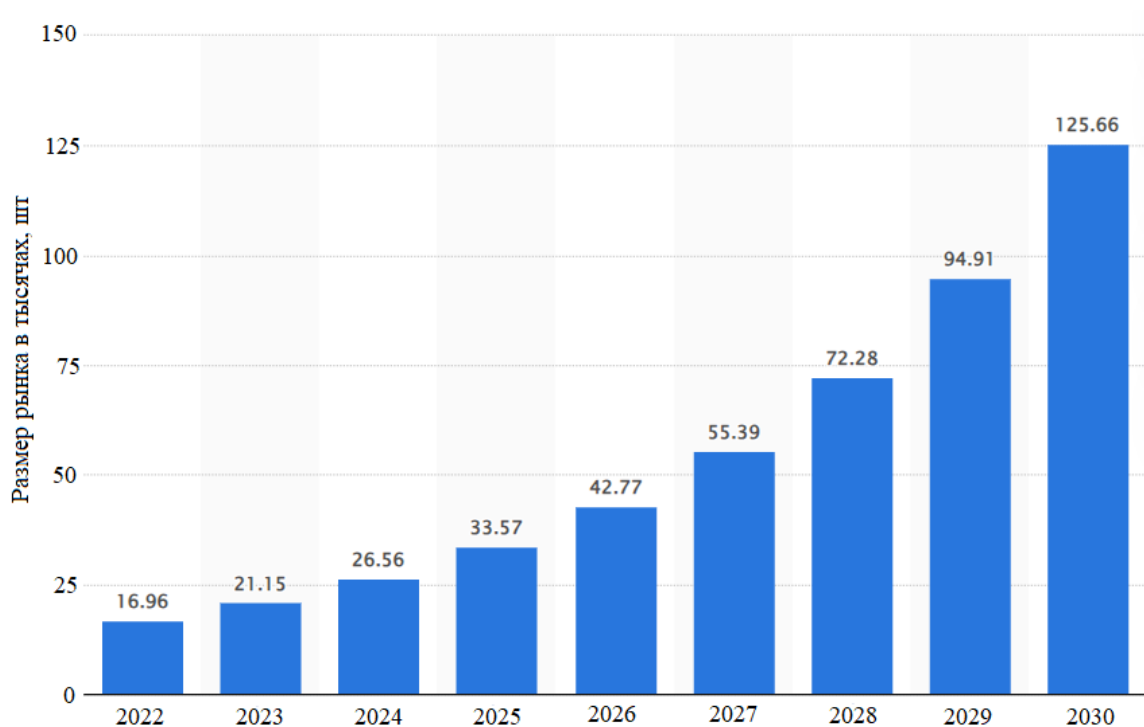


Рисунок 1.4 – Количество беспилотных автомобилей в мире в период с 2022 по прогнозный 2030 год

Уже сейчас рынок беспилотных автомобилей на 2023 год составил около 200 миллиардов долларов (рисунок 1.5) и по прогнозам Statista в ближайшие годы будет продолжать показывать значительный рост и достигнет 2,2 триллиона долларов к 2030 году.

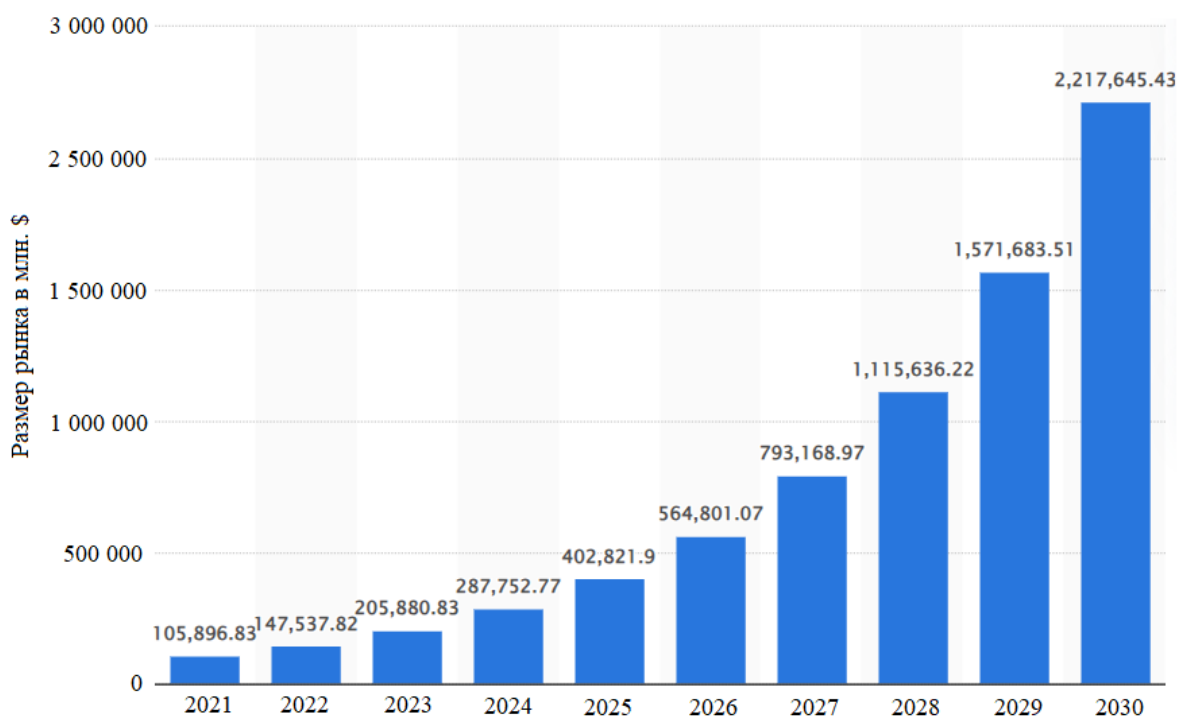


Рисунок 1.5 – Размер мирового рынка беспилотного транспорта в период с 2021 по прогнозный 2030 год

1.2. Состояние и перспективы развития автомобильной отрасли в России

Согласно распоряжениям правительства №2290-р от 23 августа 2021 года и № 4261-р от 28 декабря 2022 года определяются ключевые задачи и этапы развития данного сектора до 2030 года.

Одной из основных целей является развитие производства и использования электрических автомобилей на территории России. Для достижения этой цели планируется увеличить производственную базу для выпуска электротранспортных средств, углубить локализацию производства, а также создать необходимую инженерную и транспортную инфраструктуру.

На данный момент в Российской Федерации уже сформирован сектор разработки и производства электротранспортных средств. Несколько компаний, включая "КАМАЗ", "ГАЗ" и "Волгабас", занимаются производством электробусов, а также созданием легких коммерческих электромобилей. Представленные модели, такие как "ГАЗelle e-NN" и автомобили от "Zetta", демонстрируют рост интереса к электрическому транспорту на российском рынке.

В июле 2023 года начался этап предварительных продаж электромобиля «Атом», разработанного акционерным обществом «Кама». В течение того же года компания провела целый комплекс испытаний ключевых компонентов автомобиля, включая аккумуляторную батарею, созданную в сотрудничестве с госкорпорацией «Росатом», а также тестирования подвески, рулевой системы и шасси. Данный проект вызвал значительный интерес на рынке: в период с 2021 по 2023 годы в него были вложены инвестиции на сумму порядка 24 млрд рублей, а количество предзаказов превысило 105 тысяч единиц, как от клиентов B2C, так и B2B сегментов.

В свою очередь, «АвтоВАЗ» планирует выйти на рынок электромобилей в первой половине 2024 года, с запуском серийного производства модели «Lada e-Largus». Этот электромобиль будет представлен в нескольких модификациях, включая варианты с пятиместным и семиместным салоном, обладающие запасом хода от 350 до 400 км. Помимо этого, будет выпускаться версия грузового фургона с запасом хода 250 км, что расширит сферу его применения в коммерческом секторе.

Также существуют инициативы по развитию инфраструктуры зарядных станций. Планируется создание 144 тыс. зарядных станций к 2030 году, чтобы обеспечить комфортное использование электромобилей и удовлетворить спрос на электрозарядные станции.

Согласно статистике, «Research HUB» для компании АО «СПГ», в России в марочной структуре за февраль 2024 г. лидировал китайский автомобиль «Zeekr» с долей 51% от всех проданных электромобилей (972 шт.). За ним следовали «Evolute» (167), «Volkswagen» (154), «Москвич» (107), «Tesla» (67). По итогам 2023 года в России было реализовано 14 тыс. новых электромобилей (см. рисунок 1.6), при этом лидером продаж стал Китай (рисунок 1.7).

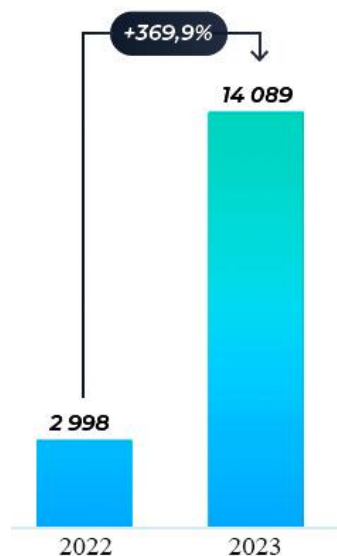


Рисунок 1.6 – Рост продаж электромобилей в России с 2022 по 2023

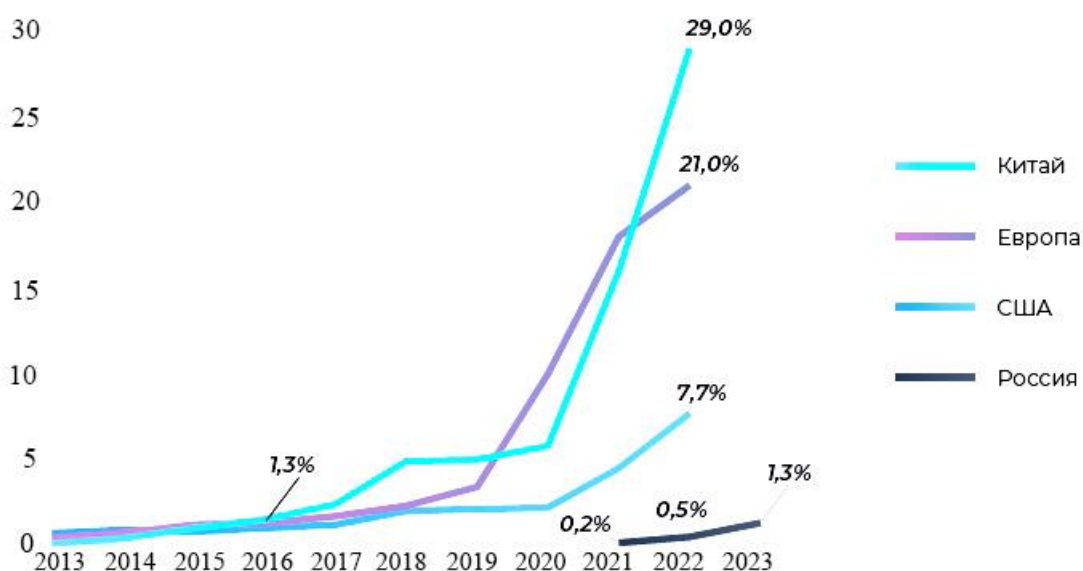


Рисунок 1.7 – Доля стран в продаже новых электромобилей в России

Согласно данным «Research HUB», за период с 2015 по 2023 годы количество зарегистрированных в Российской Федерации электромобилей возросло на 134,8%, достигнув 37,8 тыс. единиц в 2023 году (см. рисунок 1.8). Для сравнения, по информации Министерства внутренних дел (МВД), на конец 2022 года в стране было зарегистрировано около 64,5 млн транспортных средств всех типов. Следовательно, доля электромобилей составляет всего 0,037% от общего объема автопарка страны, что указывает на значительный потенциал для роста.



Рисунок 1.8 – Количество зарегистрированных в России ЭМ

Стратегические программы и планы по развитию электротранспорта в РФ

Как уже упоминалось ранее, 23 августа 2021 г. была утверждена Концепция по развитию электротранспорта в РФ на период до 2030 г., реализуемая в два этапа. К концу первого этапа планируется выпустить не менее 25 тыс. электромобилей в 2024 г. К 2030 г. производство новых ЭМ возрастет до 217 тыс. в год (см. рисунок 1.9). К этому сроку должно быть построено 8 573 единицы медленных зарядных станций, быстрых – 5 715 единиц (в 2022 г. – 1 178 и 528 единиц соответственно).

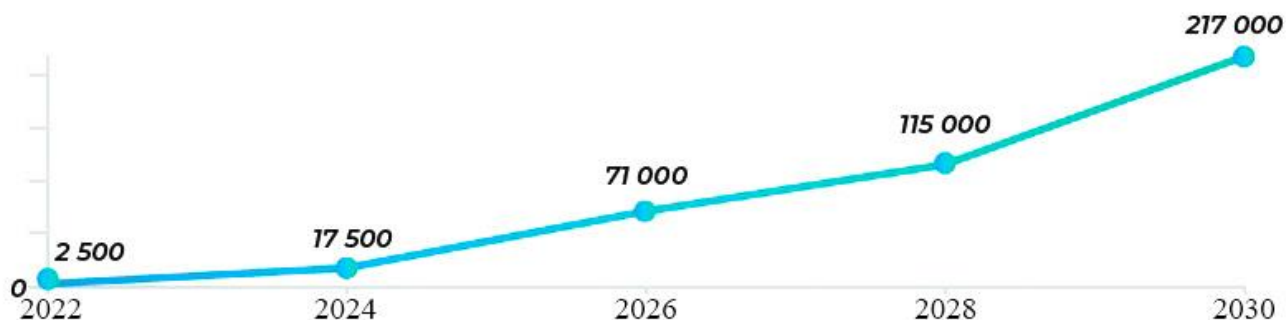


Рисунок 1.9 – Планы на производство ЭМ в России

В рамках реализации Концепции развития электротранспорта в Российской Федерации предусмотрена поддержка строительства инфраструктуры для электрических зарядных станций (ЭЗС). В настоящее время субсидии на установку одной быстрой зарядной станции составляют 2,7 млн рублей, из которых 1,8 млн рублей выделяются на приобретение оборудования, а оставшиеся 900 тыс. рублей – на технологическое присоединение к электросетям.

Второе стратегическое направление ориентировано на стимулирование спроса на частный и корпоративный электротранспорт. Для этого разработан комплекс мер, включающий:

- предоставление 25%-й скидки на приобретение отечественного электромобиля в рамках программ льготного автокредитования и автолизинга;
- субсидирование поездок в службах такси и каршеринга, что способствует снижению стоимости услуг для пользователей;
- снижение стоимости обязательного страхования гражданской ответственности владельцев транспортных средств (ОСАГО) для электромобилей;
- максимальная субсидия на покупку электромобиля через льготные программы в 2023 году составляет 625 тыс. рублей.

Параллельно реализуется федеральная программа комплексного развития городского электротранспорта, в рамках которой планируется поставка электробусов в 10 российских регионов, включая Липецкую, Ярославскую, Нижегородскую, Курскую, Саратовскую, Волгоградскую, Ростовскую области, а также Краснодарский, Красноярский и Пермский края. До 2028 года планируется поставить более 250 электробусов и установить свыше 100 зарядных станций.

Ожидается, что 70% работ по обновлению городского электрического транспорта будет выполнено в 2024 году, при этом проект реализуется на основе концессионных соглашений. Общая стоимость программы превышает 250 млрд рублей, из которых 86,5 млрд рублей будут выделены из федерального бюджета. Финансирование проекта также осуществляется за счет средств региональных бюджетов, частных инвесторов и кредитов ВЭБ.РФ.

1.3. Анализ существующих вариантов компоновки тяговых электродвигателей электромобиля

Возможные варианты компоновки тягового электропривода электроавтомобиля наиболее полно представлены в исследованиях Ren Q и Crolla D.A [102]. В работе были исследованы следующие варианты компоновок

электропривода с точки зрения технической реализации, технических характеристик и ценообразования. Были рассмотрены следующие варианты:

- Один двигатель. Задние колеса ведущие (задний привод).
- Один двигатель. Ведущие передние колеса (передний привод).
- Один двигатель. Все колеса ведущие (полный привод).
- Два двигателя с 2-мя независимыми передними и задними приводами (полный привод).
- Два независимых двигателя сзади (задний привод).
- Два независимых двигателя спереди (передний привод).
- Три двигателя. Один двигатель на переднем валу и два мотор-колеса сзади (полный привод).
- Три двигателя. Один двигатель на заднем валу и два мотор-колеса спереди (полный привод).
- 4 двигателя (независимый полный привод).

Согласно исследованиям [38, 88, 102] были определены наиболее перспективные варианты компоновки тяговых электродвигателей для электротранспорта:

- два независимых двигателя сзади (рисунок 1.10), либо спереди;
- полноприводный автомобиль с 4-мя независимыми двигателями (рисунок 1.11).

Данные компоновки позволяют осуществлять рациональное управление электромобилем при совершении манёвров и поворотов [24, 39, 56].

В данной работе будет рассмотрена и проанализирована компоновка с двумя независимыми двигателями сзади, которая позволяет избавиться от механического дифференциала и сложной системы подвески (независимая и полунезависимая) [87, 111, 137].

В таблице 1.1 приведён сравнительный анализ компоновок ведущего привода у современных и концептуальных электротранспортных средств различного назначения. В таблице указан производитель, марка, тип транспорта (ЛА – легковой автомобиль, Г – грузовой, МА – микроавтобус, А – автобус, СА –

спортивный автомобиль, КБ – концепт беспилотного автомобиля), тип применяемого двигателя и тип привода, где PMSM – это Permanent Magnet Synchronous Motor, синхронный двигатель с постоянными магнитами.

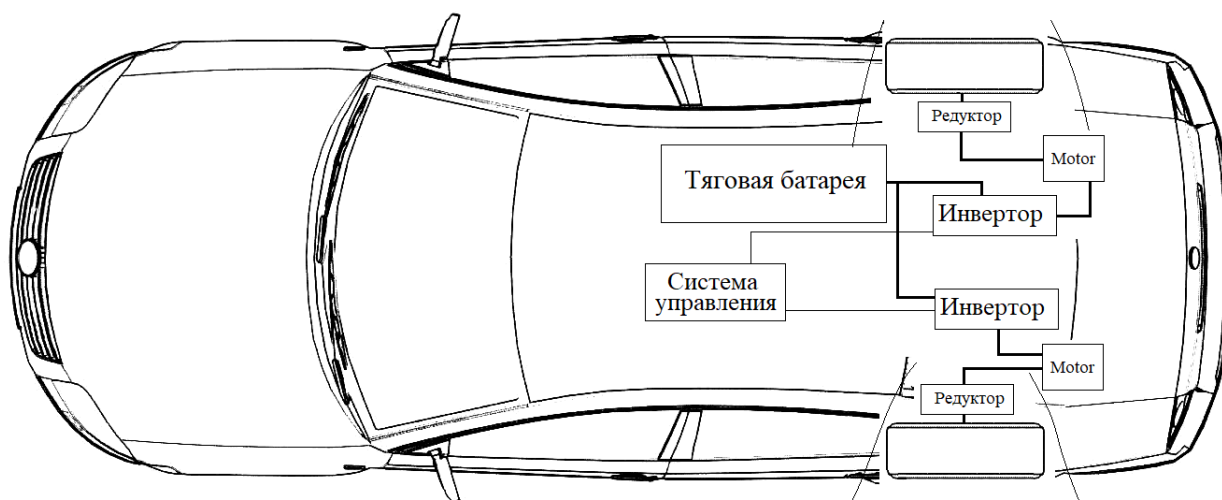


Рисунок 1.10 – Компоновка трансмиссии с 2-мя независимыми двигателями сзади

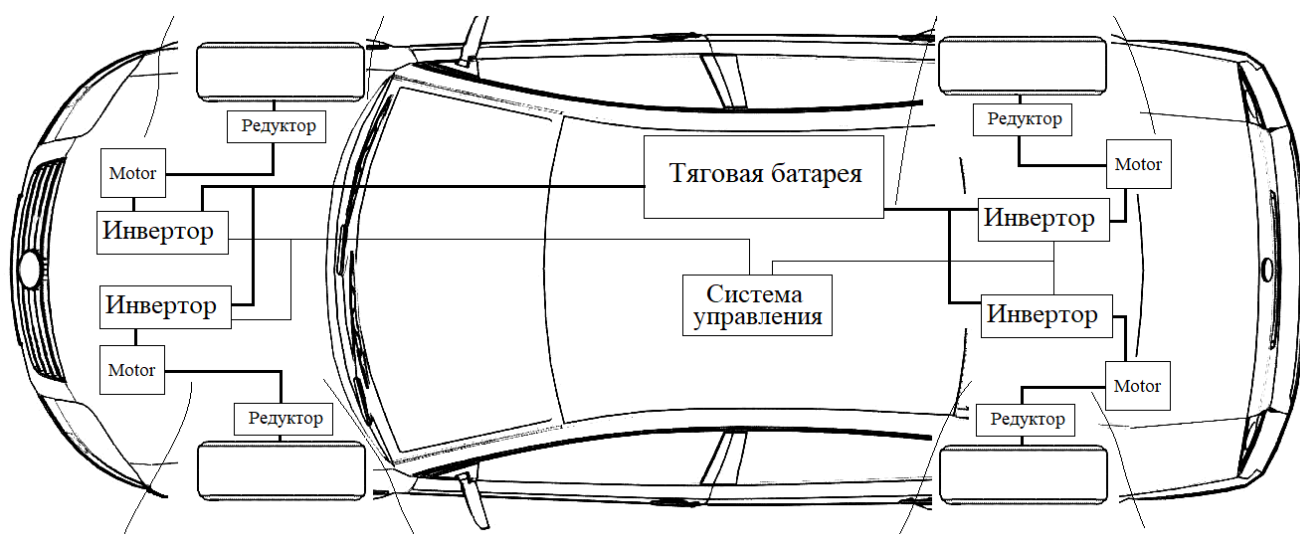


Рисунок 1.11 – Компоновка полноприводного автомобиля с 4-мя независимыми двигателями

Таблица 1.1 – Компоновка тягового электропривода электротранспорта

Производитель	Марка	Тип	Тип двигателя	Количество двигателей	Тип привода
Nissan	LEAF	ЛА	PMSM	1	Передний
Tesla	Model 3	ЛА	PMSM	1	Задний
Li Auto	L6	ЛА	PMSM	2	Полный
BYD	ATTO 3	ЛА	PMSM	1	Передний

GAC	Trumpchi GE3	ЛА	PMSM	1	Передний
Nio	EC6	ЛА	Асинхронный	2	Полный
Byd	Atto 3	ЛА	PMSM	1	Передний
Xiaomi	SU7	ЛА	PMSM	1	Задний
Audi	Q4 e-tron	ЛА	PMSM	2	Полный
BMW	iX	ЛА	PMSM	2	Полный
МАЗ	4381EE	Г	Асинхронный	1	Задний
JMC	E820	Г	PMSM	1	Задний
Volvo	FMX Electric	Г	Асинхронный	2/3	Задний / Полный
Газель	Next Electro	МА	Асинхронный	1	Задний
Nissan	e-NV200 Voltia	МА	PMSM	1	Передний
КАМАЗ	6282	А	Асинхронный	2	Задний
ЛиАЗ	6274	А	Асинхронный	2	Задний
Volgabus	5270.E0	А	Асинхронный	1	Задний
Lotus	Emeya	СА	PMSM	2	Полный
Porsche	Taycan	СА	PMSM	2	Полный
Rimac	Concept Two	СА	PMSM	4	Полный
Audi	Aicon	КБ	PMSM	4	Полный
Mercedes-Benz	VISION AVTR	КБ	-	4	Полный
Lexus	LF-30 Concept	КБ	-	4	Полный

Анализ таблицы 1.1 показывает, что в настоящее время основные производители электромобилей (США, Китай, Япония, Германия) выпускают электромобили с компоновкой 2 и 4 тяговых электродвигателя.

1.4. Анализ эффективности существующих систем управления электродвигателем при поворотах и манёврах

Для управления скоростью нескольких тяговых двигателей электромобилей при осуществлении поворотов и манёвров используется система электронного дифференциала (СЭД). При движении электромобиля при повороте СЭД регулирует скорость вращения электродвигателей таким образом, что скорость внутреннего колеса уменьшается, а внешнего увеличивается (рисунок 1.12). Такой

алгоритм управления позволяет улучшить устойчивость и управляемость электроавтомобилем при поворотах [62].

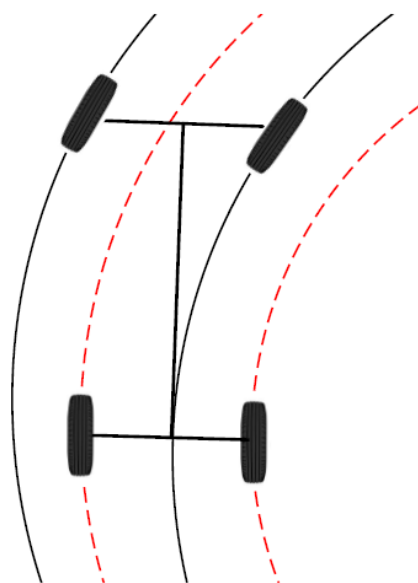


Рисунок 1.12 – Траектория движения колес при повороте электроавтомобиля [7]

Использование САД вместо механического дифференциала имеет ряд преимуществ – улучшение управляемости, уменьшение конечного веса и стоимости электроавтомобиля [1, 3, 4, 6, 8, 45, 134-136]. В таблице 1.2 приведены сравнения механического и системы электронного дифференциала.

Таблица 1.2 – Сравнение механического и электронного дифференциала

Тип дифференциала	Плюсы	Минусы
Механический	1. Простота реализации.	1. Большая масса по сравнению с электронным 2. Меньшая надёжность из-за наличия трущихся частей 3. Необходимость использовать механизм блокировки, чтобы автомобиль мог двигаться по льду, и по сложным дорогам. 4. Крутящий момент ограничивается колесом с наименьшим сцеплением.

Электронный	<ol style="list-style-type: none"> 1. Отсутствие механических элементов и как следствие уменьшение массы и повышение надёжности 2. Повышение манёвренности 3. Экономия подпольного пространства автомобиля 4. Реконфигурируемость (задание новой модели поведения без изменения конструктива) 5. Крутящий момент не ограничивается колесом с наименьшим сцеплением 6. Точное задание скорости вращения (что позволяет реализовать дополнительные системы помощи вождения) 7. Уменьшение конечной стоимости электроавтомобиля 8. Простая интеграция во вспомогательные системы беспилотных автомобилей 9. Увеличение «пробега» на одном заряде благодаря снижению конечной массы 	<ol style="list-style-type: none"> 1. Сложность в реализации на основе существующих систем электронного дифференциала.
-------------	--	---

При использовании электронного дифференциала и двух двигателей в режиме прямого привода повышается КПД и характеристики управления автомобилем. Достигается это за счёт того, что уменьшается количество ступеней передачи мощности, а значит мы получаем снижение потерь [28].

1.4.1. Линейная система управления электроприводом

В основе функционирования линейной системы, схема которой показана на рисунке 1.13, лежит измерение угла поворота руля с помощью переменного резистора, который выполняет роль датчика поворота. С данного резистора

формируется задающее значение сигнала рассогласования между электродвигателями, который поступает на блок управления. Блок управления состоит из операционных усилителей, обеспечивающих обработку и усиление сигнала рассогласования, необходимого для передачи управляющих воздействий на исполнительные механизмы. На выходе блока управления формируются два независимых сигнала скорости, которые поступают на два силовых преобразователя, каждый из которых регулирует скорость вращения соответствующего электродвигателя, что позволяет создать разницу в скоростях вращения колес, необходимую для выполнения манёвров.

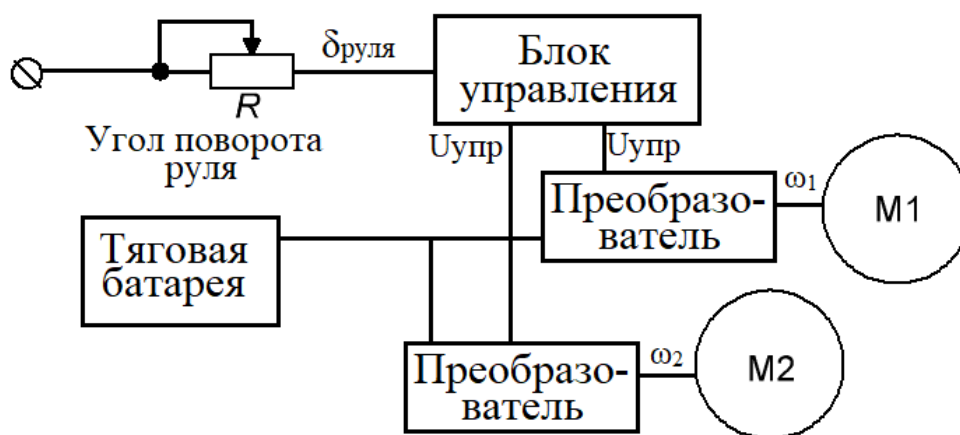


Рисунок 1.13 – Схема линейной системы управления электродвигателями

На основе анализа данной схемы (рисунок 1.13) можно заключить, что линейная схема системы электронного дифференциала отличается относительной простотой конструкции. Недостатком данной схемы является отсутствие учёта ряда факторов при поворотах электроавтомобиля (конструктивных особенностей транспортного средства, текущей скорости движения).

1.4.2. Нелинейная система управления электроприводом

Одно из первых упоминаний нелинейной системы электронного дифференциала для электроавтомобиля было в работе [76]. Идея электронного дифференциала данного типа появилась в конце XX века с появлением электроавтомобилей и гибридных транспортных средств. Однако первые патенты, связанные с этой системой, начали появляться в 1980-х и 1990-х годах. Один из первых патентов (DE10341787B3, 2003 г.) на нелинейный электронный

дифференциал был зарегистрирован немецкой компанией Daimler AG. Позже другие автопроизводители, такие как Tesla (US20210310551A1, 2021 г.), BMW и другие (CN101758854B, 2010 г.) начали разрабатывать свои версии электронных дифференциалов для электромобилей [29, 99].

Нелинейный электронный дифференциал был создан на основе существующих технологий управления скоростью вращения и рулевым управлением, а его разработка связана с прогрессом в области электроники, электродвигателей и систем управления.

Иллюстрация работы нелинейной СЭД при движении транспортного средства с поворотом налево представлена на рисунке 1.14, где L – длина колесной базы автомобиля, d – ширина колесной базы. Для определения угла поворота транспортного средства используется датчик положения рулевого колеса. При движении автомобиля в прямом направлении угол поворота ($\delta = 0$). При повороте автомобиля ($\delta \neq 0$) активируется система СЭД [95, 120, 132], в случае, когда $\delta > 0$ передние колеса поворачиваются влево, при $\delta < 0$, осуществляется поворот вправо [77].

Структурная схема работы нелинейной системы электронного дифференциала представлена на рисунке 1.15. Принцип работы системы в целом аналогичен работе линейной системы, описанной ранее. Однако, в данном случае учитывается измерение скорости автомобиля и ряд конструктивных постоянных транспортного средства (L , K , d_r , r , l_r). В качестве управляющего элемента используется микроконтроллер, заменяющий блок управления. Это обусловлено тем, что расчёт величины скорости выполняется с использованием математического аппарата, согласно формулам (1.1–1.5) [135]. После вычисления, соответствующие значения скорости вращения привода, данные значения передаются на силовые преобразователи, а затем на тяговые электродвигатели, что обеспечивает более точное управление движением автомобиля по сравнению с линейной системой.

Внутренний угол поворота переднего колеса определяется по формуле:

$$\delta_1 = \arctan \left[\frac{L \cdot \tan(\delta)}{L - \left(\frac{K}{2} \cdot \tan(\delta)\right)} \right] \quad (1.1)$$

где K – расстояние между левым и правым шкворнем;

L – расстояние между передним и задним колесом;

δ – угол поворота.

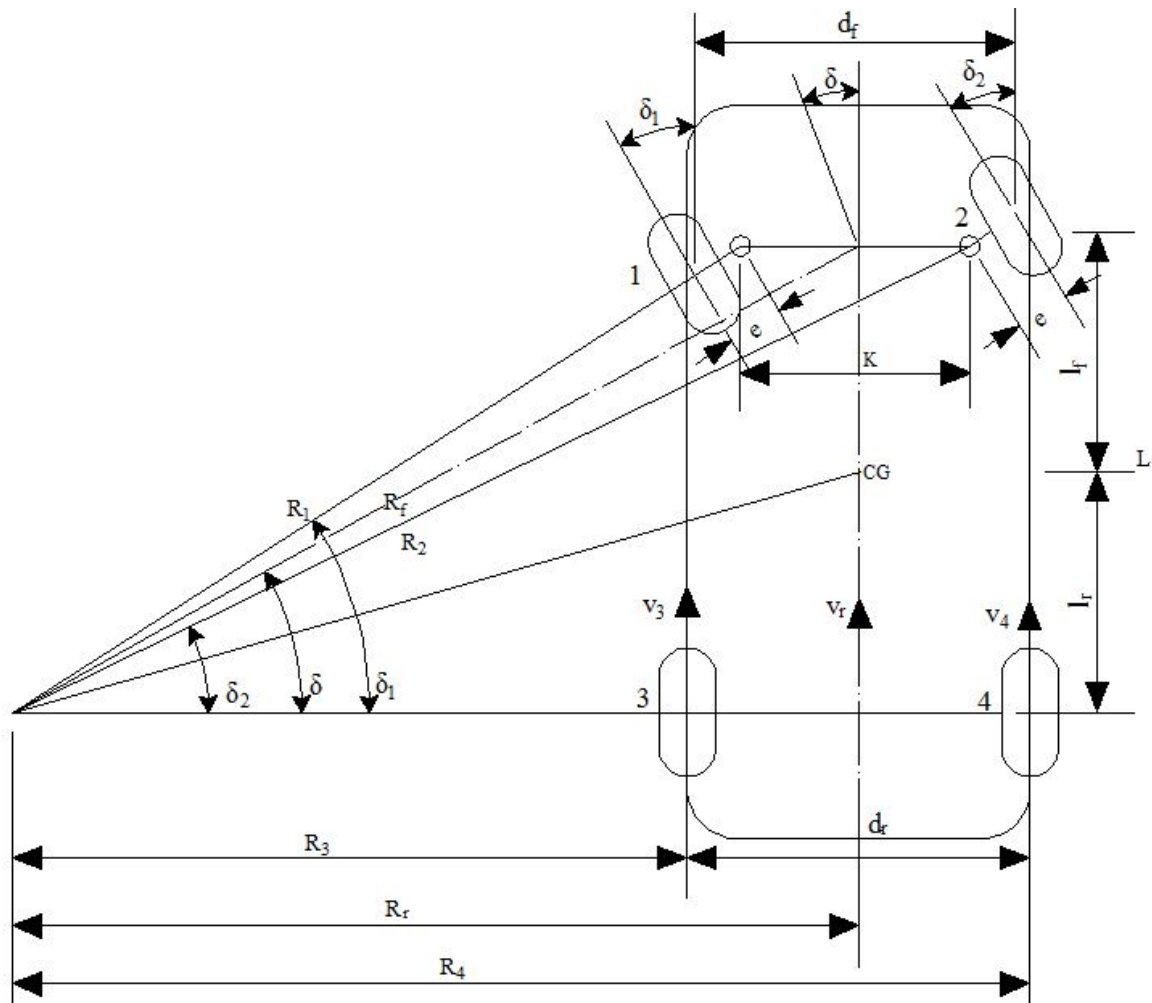


Рисунок 1.14 – Схема траектории движения автомобиля на повороте [135]

Внешний угол поворота переднего колеса определяется по формуле:

$$\delta_2 = \arctan \left[\frac{L \cdot \tan(\delta)}{L + \left(\frac{K}{2} \cdot \tan(\delta)\right)} \right] \quad (1.2)$$

Радиусы поворота передних внутренних и внешних колес, задних внутренних и внешних колес могут быть соответственно выражены как:

$$R_1 = \frac{L}{\sin(\delta_1)} \quad (1.3)$$

$$R_2 = \frac{L}{\sin(\delta_2)}$$

$$R_3 = \frac{L}{\tan(\delta)} - \frac{d_r}{2}$$

$$R_4 = \frac{L}{\tan(\delta)} + \frac{d_r}{2}$$

где d_r – расстояние между задними колесами.

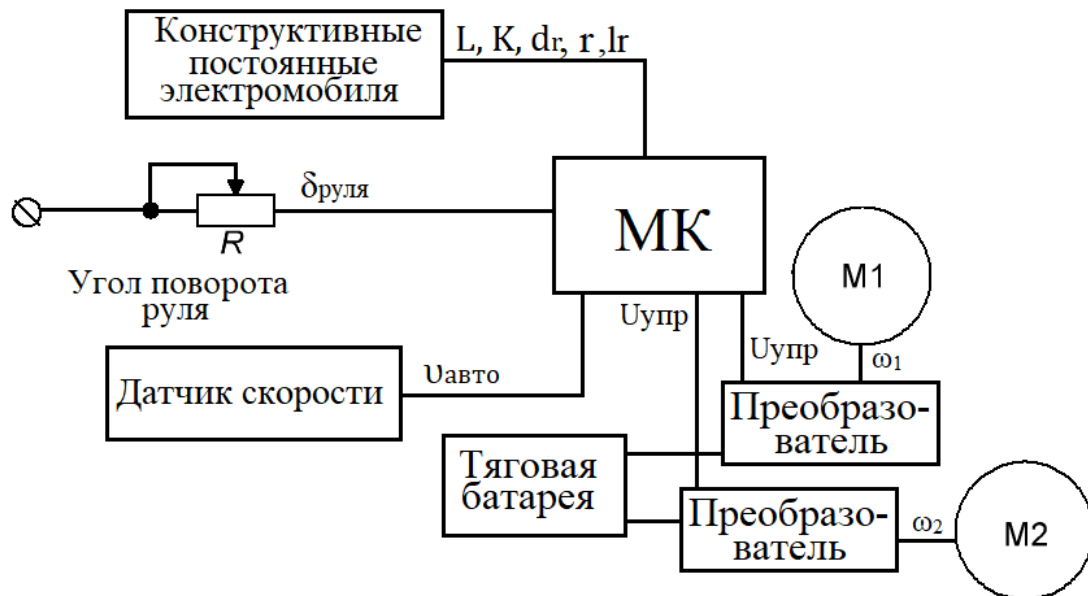


Рисунок 1.15 – Структурная схема работы нелинейной системы электронного дифференциала

Радиус центра тяжести электромобиля определяется по выражению:

$$R_{cg} = \sqrt{R_3 + \left(\frac{d_r}{2}\right)^2 + l_r^2} \quad (1.4)$$

где l_r – расстояние между задним колесом и центром тяжести.

Тогда, угловые скорости передних внутренних и внешних колес, а также задних внутренних и внешних колес могут быть соответственно выражены следующим образом:

$$\omega_1 = \frac{v \cdot R_1}{R_{cg} \cdot r} \quad (1.5)$$

$$\omega_2 = \frac{v \cdot R_2}{R_{cg} \cdot r}$$

$$\omega_3 = \frac{v \cdot R_3}{R_{cg} \cdot r}$$

$$\omega_4 = \frac{v \cdot R_4}{R_{cg} \cdot r}$$

где r – радиус колеса;

v – скорость автомобиля.

Анализ выражений (1.1–1.5) показывает, что данная модель электронного дифференциала не учитывает следующие параметры: состояние шин и динамические характеристики автомобиля, влияние внешних погодных условий, стиль вождения и др.

Выводы по главе 1

1. Развитие автомобильной отрасли в мире и России идёт в направлении увеличения производства электроавтомобилей.

2. Наиболее перспективными для будущего электротранспорта можно выделить 2 варианта компоновки тяговых электродвигателей:

-два независимых электродвигателя сзади (спереди);

-полноприводный автомобиль с 4-мя независимыми электродвигателями на каждое колесо.

3. Использование СЭД вместо механического дифференциала имеет ряд преимуществ – улучшение управляемости, уменьшение конечного веса и стоимости электроавтомобиля.

4. Недостатком линейной системы электронного дифференциала является отсутствие учёта ряда факторов при поворотах электроавтомобиля (конструктивных особенностей транспортного средства, текущей скорости движения).

5. Нелинейная система электронного дифференциала показывает более высокую эффективность управления по сравнению с линейной системой, поскольку учитывает не только угол поворота, но и скорость электроавтомобиля, а также его конструктивные особенности.

ГЛАВА 2. ТЕОРЕТИЧЕСКИЕ АСПЕКТЫ РАЗРАБОТКИ СИСТЕМЫ ЭЛЕКТРОННОГО ДИФФЕРЕНЦИАЛА ДЛЯ УПРАВЛЕНИЯ ЭЛЕКТРОАВТОМОБИЛЕМ НА ОСНОВЕ НЕЙРОННОЙ СЕТИ

Для анализа эффективности работы линейной и нелинейной систем электронного дифференциала разработаны математические модели данных систем в среде MatLab/Simulink.

2.1. Математические модели линейной и нелинейной систем управления

Линейная СЭД

Математическая модель линейной СЭД в среде MatLab/Simulink приведена на рисунке 2.1, результат вычисления управляющего ШИМ-сигнала при соотношении плеч переменного резистора $1,5R / 1R$, приведён на рисунке 2.2.

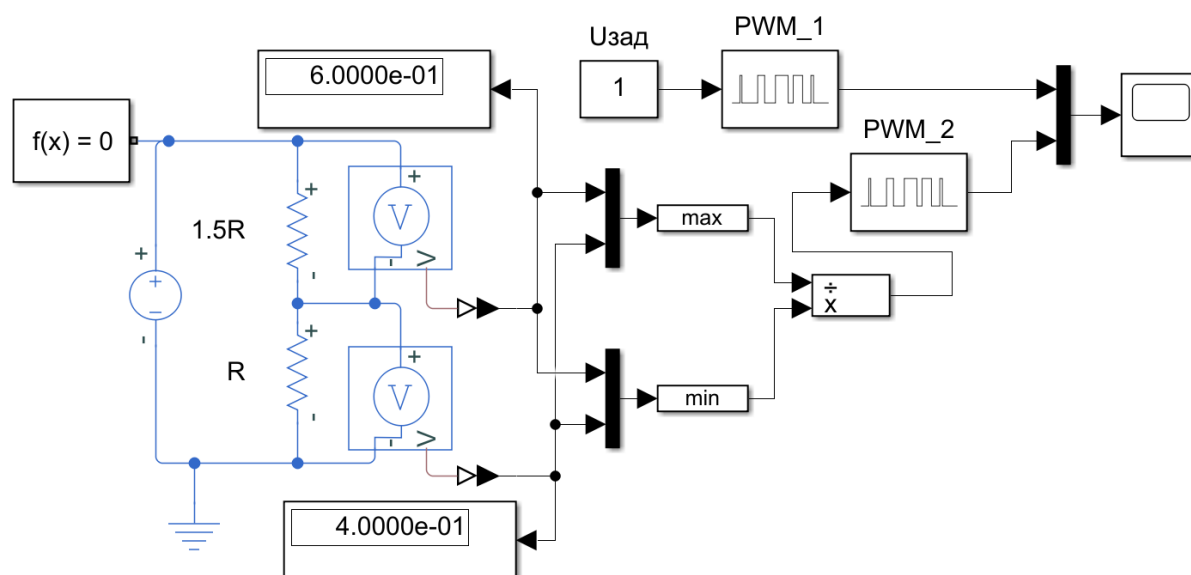


Рисунок 2.1 – Математическая модель линейной СЭД

На основе полученных результатов в первой главе и моделирования в среде MatLab/Simulink можно выделить следующие выводы:

Преимущества системы:

1. Простота реализации.

Недостатки системы:

1. Снижение качества управления электроприводом на малых и высоких скоростях, а также небольших углах поворота автомобиля.

2. Не учитываются конструктивные особенности автомобиля, включая распределение массы и геометрию колесной базы.
3. Не учитываются динамические факторы при движении электромобиля (силы инерции, сопротивление воздуха и т.д.).
4. Отсутствие учёта внешних условий эксплуатации, влияющих на качество управления (погодные и дорожные факторы).
5. Отсутствие возможности адаптации под различные стили вождения и динамические режимы эксплуатации (спортивный или экономичный).
6. Не учитывается техническое состояние автомобиля (износ шин, весовые характеристики, люфты механизмов).
7. Отсутствие возможности интеграции с системами безопасности и помощи водителю (ABS, ESP и круиз-контроль).

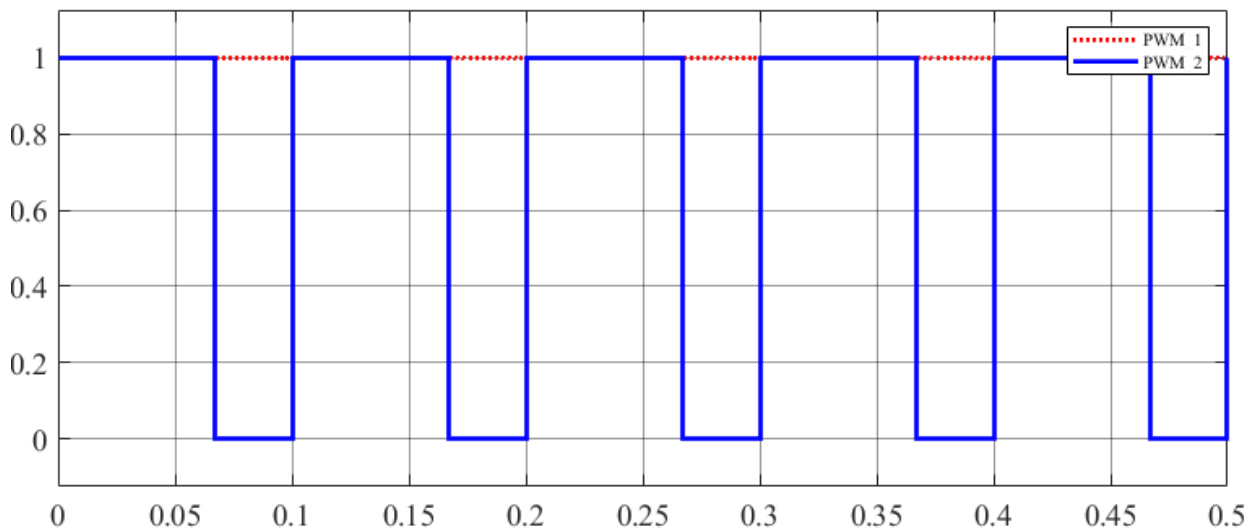


Рисунок 2.2 – Управляющий ШИМ-сигнал при соотношении плеч переменного резистора $1,5R / 1R$

Нелинейная СЭД

Математическая модель нелинейной СЭД в среде MatLab/Simulink и результат расчёта скорости вращения 2-х задних колёс при угле поворота 10^0 приведен на рисунке 2.3, где $\omega_{авт}$ – это значение скорости вращения колёс при движении прямо, взятое в качестве сравнительного образца; ω_3 , ω_4 – скорость вращения заднего левого и правого колёс соответственно; $V_{авт}$ – значение линейной скорости движения автомобиля; L , d_r , I_r , r – конструктивные постоянные автомобиля (см.

рисунок 1.14). Результат расчёта скорости вращения 2-х задних колёс при угле поворота 30° приведен на рисунке 2.4.

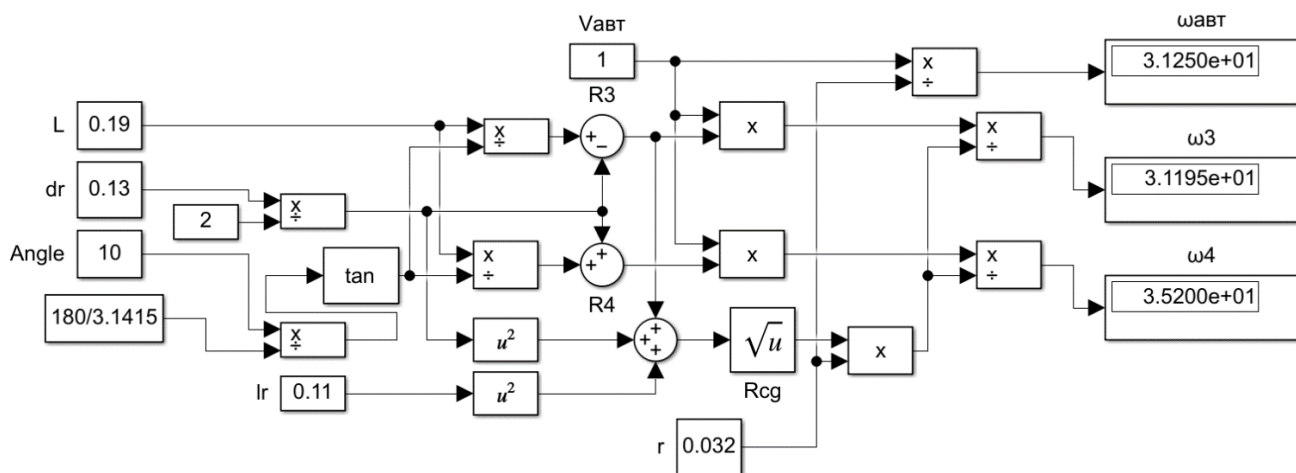


Рисунок 2.3 – Математическая модель нелинейной СЭД при угле поворота 10°

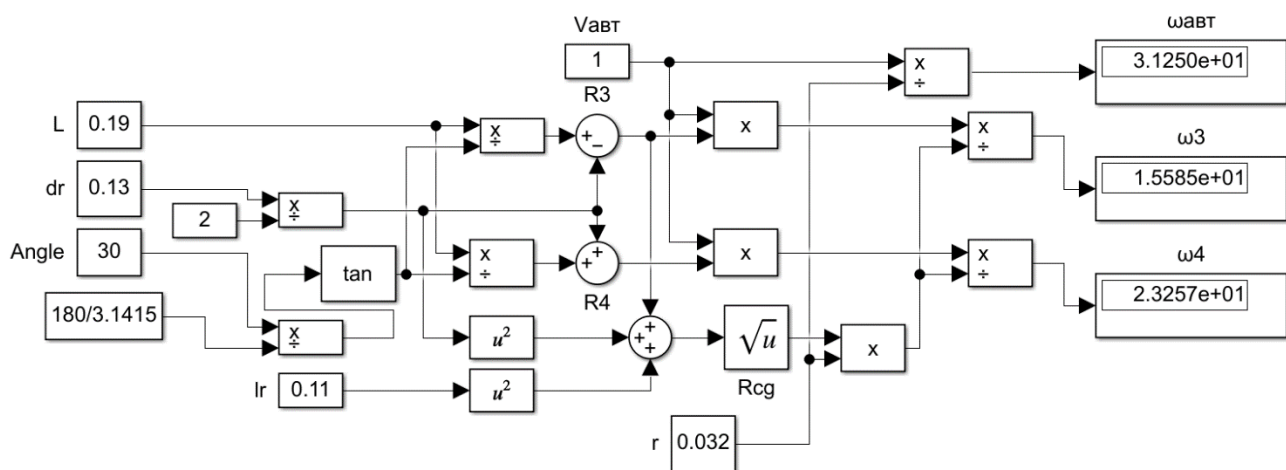


Рисунок 2.4 – Математическая модель нелинейной СЭД при угле поворота 30°

На основе полученных результатов в первой главе и моделирования в среде MatLab/Simulink можно выделить следующие выводы:

Преимущества системы:

1. Относительная простота реализации.
2. Частичный учёт конструктивных особенностей автомобиля.
3. Возможность применения в полноприводных наземных транспортных средствах.

Недостатки системы:

1. При скорости выше 50 км/ч наблюдается снижение качества управления, что ограничивает применение системы на высоких скоростях [135].

2. Отсутствие учёта внешних условий эксплуатации, влияющих на качество управления (погодные и дорожные факторы).

3. Отсутствие возможности адаптации под различные стили вождения и динамические режимы эксплуатации (спортивный или экономичный).

4. Не учитывается техническое состояние автомобиля (износ шин, весовые характеристики, люфты механизмов).

5. Отсутствие возможности интеграции с системами безопасности и помощи водителю (ABS, ESP и круиз-контроль).

2.2. Влияние инерционных сил на характер движения транспортного средства при поворотах и манёврах

Анализ научно-технической литературы и математических моделей линейной и нелинейной систем показал, что при скорости электроавтомобиля выше 50 км/ч данные системы не учитывают негативное влияние на управляемость возрастающих инерционных сил и аэродинамического сопротивления воздушной среды.

В частности, для более полного описания поведения автомобиля при поворотах и манёврах требуется учесть продольные и поперечные силы, силы инерции и взаимодействие каждого колеса с дорогой. На рисунке 2.5 представлена динамическая модель полноприводного электромобиля, включающая семь степеней свободы, которые учитывают движение автомобиля и позволяют более точно описать его поведение [23, 80, 91, 141]. Данные степени свободы включают продольные движения (вдоль оси X), поперечные движения (вдоль оси Y), рысканье (вращение вокруг оси Z) и динамику вращения каждого из четырёх колёс. В модель также включены следующие упрощения:

1. Центр динамической системы координат находится в том же месте, что и центр тяжести транспортного средства;

2. Независимо от роли подвески, автомобиль движется в положении, параллельном земле;

3. Влияние системы рулевого управления игнорируется за счёт отсутствия люфтов в механизмах;

4. Физические свойства всех четырех колёс одинаковы.

На рисунке 2.5 l_f представляет собой расстояние от передней оси до центра масс транспортного средства; l_r – расстояние от задней оси до центра масс транспортного средства; F_{xi} и F_{yi} – продольные и поперечные силы воздействия на колёса соответственно; γ – скорость рыскания; d_f и d_r передняя и задняя колея соответственно; β – угол бокового увода электромобиля; δ – угол поворота передних колес; V_x – скорость транспортного средства по оси X (продольная); V_y – скорость транспортного средства по оси Y (поперечная). Нижний индекс j указывает левое (l) колесо или правое (r), а i указывает на переднее (f) колесо или заднее (r).

Продольное движение вдоль оси X описывается следующим образом [34]:

$$m \left(V_x' + V_y \frac{d\gamma}{dt} \right) = F_{x3} + F_{x4} + (F_{x1} + F_{x2}) \cdot \cos(\delta) - (F_{y1} + F_{y2}) \cdot \sin(\delta) - (F_{z1}f_1 + F_{z2}f_2) \cdot \cos(\delta) - F_{z3}f_3 - F_{z4}f_4 - \frac{C_D A}{21,15} V_x^2 \quad (2.1)$$

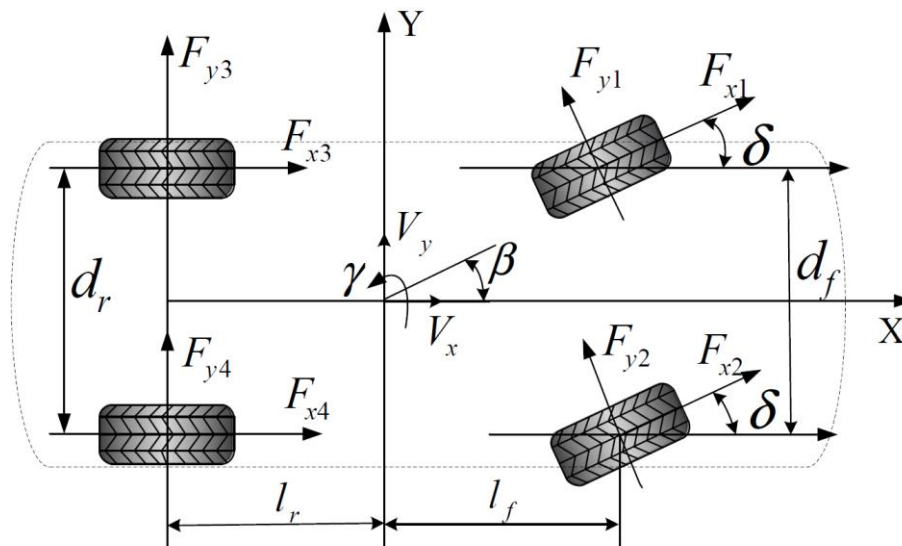


Рисунок 2.5 – Силы, действующие на автомобиль во время поворота [91]

Боковое движение вдоль оси Y описывается при помощи выражения:

$$m \left(V_y' + V_x \frac{d\gamma}{dt} \right) = F_{y3} + F_{y4} + (F_{x1} + F_{x2}) \cdot \sin(\delta) + (F_{y1} + F_{y2}) \cdot \cos(\delta) -$$

$$-(F_{z1}f_1 + F_{z2}f_2) \cdot \sin(\delta) - F_{z4}f_4 \quad (2.2)$$

Динамика рыскания определяется как:

$$\begin{aligned} J_z \gamma' = & \left(-\frac{d_f}{2} \cos(\delta) + l_f \sin(\delta) \right) \cdot F_{x1} + \left(\frac{d_f}{2} \cos(\delta) + l_f \sin(\delta) \right) \cdot F_{x2} + \\ & + \frac{d_r}{2} (F_{x4} - F_{x3}) + \frac{d_f}{2} \sin(\delta) (F_{y1} - F_{y2}) + l_f \cos(\delta) \cdot (F_{y1} + F_{y2}) - \\ & - l_r (F_{y3} + F_{y4}) \end{aligned} \quad (2.3)$$

где m – масса транспортного средства;

f_1, f_2, f_3, f_4 – коэффициенты сопротивления качению каждого из колёс;

A – наветренная область автомобиля;

J_z – момент инерции рыскания;

C_D – коэффициент сопротивления воздуха.

Динамика вращения колес транспортного средства выражается как:

$$J_i \cdot \omega'_i = T_{xi} - T_{bi} - T_{fi} - F_{xi} \cdot R_i \quad (2.4)$$

где J_i – момент инерции колеса;

ω_i – угловая скорость i -го колеса;

T_{xi} – крутящий момент i -го колеса;

T_{bi} – момент торможения i -го колеса;

T_{fi} – момент сопротивления качения i -го колеса;

R_i – радиус i -го колеса.

Момент сопротивления качения каждого колеса при повороте автомобиля выражается следующим образом:

$$T_{fi} = f \cdot F_{zi} \cdot R_i \quad (i = 1,2,3,4) \quad (2.5)$$

$$\begin{aligned} F_{z1} &= \frac{m}{l} \left(\frac{g \cdot l_r}{2} - \frac{a_x \cdot h_g}{2} - \frac{a_y \cdot h_g \cdot l_r}{d_f} \right) \\ F_{z2} &= \frac{m}{l} \left(\frac{g \cdot l_r}{2} - \frac{a_x \cdot h_g}{2} + \frac{a_y \cdot h_g \cdot l_r}{d_f} \right) \\ F_{z3} &= \frac{m}{l} \left(\frac{g \cdot l_r}{2} - \frac{a_x \cdot h_g}{2} - \frac{a_y \cdot h_g \cdot l_f}{d_f} \right) \end{aligned} \quad (2.6)$$

$$F_{z4} = \frac{m}{l} \left(\frac{g \cdot l_r}{2} - \frac{a_x \cdot h_g}{2} + \frac{a_y \cdot h_g \cdot l_f}{d_f} \right)$$

где g – ускорение свободного падения;

F_{zi} ($i = 1, 2, 3, 4$) – вертикальная динамическая нагрузка каждого i -го колеса;

a_x, a_y – продольное и поперечное ускорение транспортного средства;

h_g – высота центра тяжести транспортного средства;

l – колесная база транспортного средства.

2.3. Влияние погодных условий на эффективность управления транспортным средством

На поведение автомобиля при прохождении поворотов оказывают влияние не только такие основные параметры, как угол поворота руля, скорость движения, центростремительное ускорение и взаимодействие шин с дорожным покрытием. Важную роль играют и внешние условия, включая погодные факторы, такие как влажность, температура, обледенение и наличие осадков. Изменения в погодных условиях могут существенно снизить коэффициент сцепления шин с дорогой, увеличивая риск заноса и ухудшая управляемость. Следовательно, для обеспечения безопасного и предсказуемого поведения автомобиля необходимо учитывать как механические параметры, так и погодные факторы, которые значительно влияют на динамику движения при поворотах [9, 31].

Например, исследование [9] изучало влияние неблагоприятных погодных условий на поведение автомобилей при разных скоростях на горных и равнинных автомагистралях Вайоминга. Регрессионный анализ, основанный на данных моделирования, позволил определить, при каких условиях транспортное средство с наибольшей вероятностью отклонится от желаемой траектории. Основные выводы включают следующие пункты:

1. Запасы на занос значительно снижаются в неблагоприятных погодных условиях, особенно при высоких скоростях движения и крутых поворотах. Так, увеличение скорости на 10% приводит к снижению запаса по заносу на 15%.

2. Легковые автомобили требуют большего бокового трения по сравнению с тяжелыми грузовиками и внедорожниками.

3. Эффективность торможения и устойчивость автомобиля зависят от состояния дорожного покрытия: на заснеженных дорогах применение тормозов увеличивает вероятность заноса, особенно у грузового транспорта.

Эти результаты подчеркивают важность учета как погодных условий, так и особенностей дорожного покрытия при оценке стабильности и безопасности транспортных средств на поворотах.

В данном контексте запас по заносу обозначает предельное значение бокового ускорения или сил сцепления, при котором транспортное средство может сохранять управляемость и оставаться на желаемой траектории без перехода в скольжение (занос). Иными словами, это резерв безопасности, показывающий, насколько автомобиль ещё может ускоряться или двигаться по стабильной траектории без риска потери сцепления шин с дорогой.

При снижении запаса по заносу до нуля автомобиль достигает предела сцепления и начинает скользить или терять управляемость, что в условиях поворота особенно критично для безопасности движения.

На рисунке 2.6 представлено влияние скорости движения легковых автомобилей на поведение транспортного средства в различных дорожных условиях при прохождении поворота с умеренной кривизной. В [9] установлено, что на заснеженных дорогах запас по заносу приближался к нулю, что свидетельствует о критической важности регулировки скорости движения для поддержания автомобиля на соответствующей траектории. На более крутых поворотах и участках с большими уклонами задача сохранения контроля над автомобилем в неблагоприятных погодных условиях становилась ещё более сложной.

Стоит отметить, что создание полноценной математической модели, способной учитывать все ранее перечисленные аспекты движения автомобиля (см. разделы 2.1-2.3) на поворотах, представляет собой чрезвычайно сложную задачу. В такой модели должны быть задействованы многочисленные параметры, включая скорость, угол поворота, конструктивные характеристики автомобиля, силы

инерции, взаимодействие шин с дорожным покрытием и погодные условия, стиль вождения.

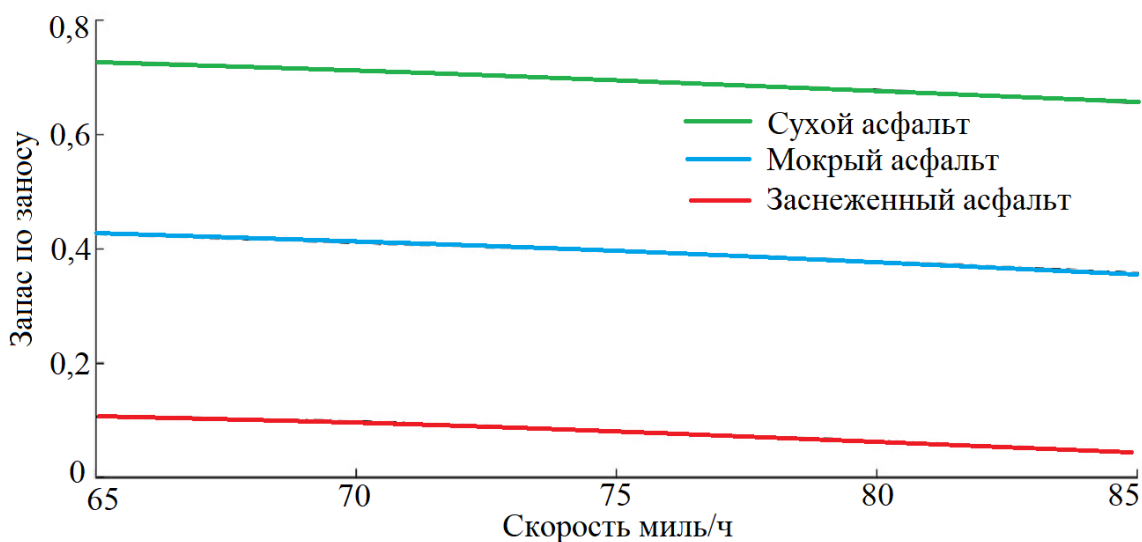


Рисунок 2.6 – Влияние скорости легкового автомобиля на запас по заносу для различных условий дорожного покрытия (умеренно крутой поворот) [9].

Высокая степень сложности и количество взаимодействующих факторов затрудняют практическую реализацию модели, которая бы учитывала все ранее перечисленные факторы. В связи с этим более рациональным подходом является использование метода «от практики» на основе искусственных нейронных сетей [15, 51, 52, 76, 133, 142]. В этом случае применение нейронных сетей, обученных на экспериментальных данных, полученных на основе реальных испытаний моделей, способны эффективно управлять скоростью электропривода электроавтомобиля, обеспечивая его устойчивость и управляемость при выполнении поворотов и манёвров с учётом соответствующих факторов.

2.4. Математическая модель системы электронного дифференциала на основе НС

Разработанная структурная схема системы электронного дифференциала для 2-х независимых двигателей сзади, показанная на рисунке 2.7. Работа СЭД заключается в следующем: НС получает значение скорости вращения задних левого и правого колёс (V_1 , V_2), а также общей скорости движения электроавтомобиля

(V^*) с установленных датчиков скорости и угла поворота руля (δ). Система также может получать иные внутренние параметры: ускорение автомобиля, данные с гироскопа, сигналы от тормозной системы, температуру тяговых двигателей и прочее, а также внешние параметры: температуру окружающей среды, влажность воздуха, расстояние до ближайшего препятствия, освещённость, время суток, учитывание характера дорожного покрытия и др.

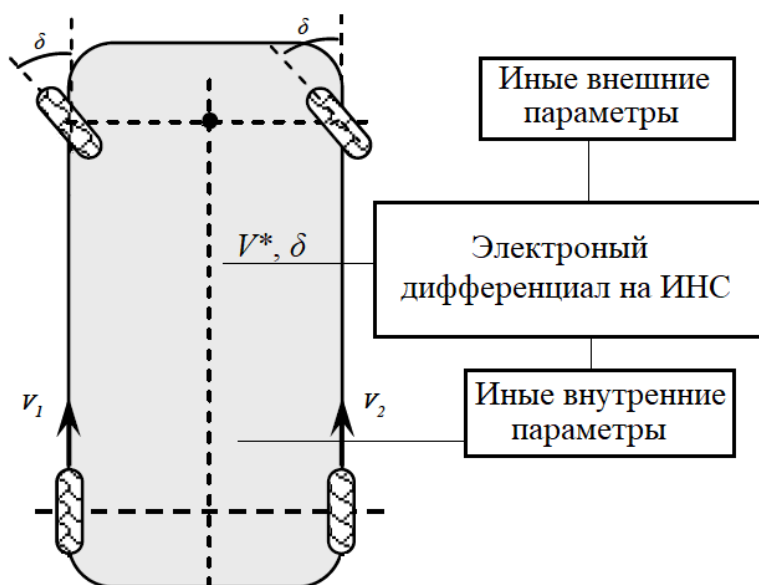


Рисунок 2.7 – Структурная схема системы электронного дифференциала на основе НС

На выходе электронного дифференциала, с учётом внутренних и внешних факторов, формируется задающее значение скорости вращения колёс, которое после передается на регулятор скорости (П, ПИ, ПИД-регулятор, регулятор, основанный на нечёткой логике и т.п.) тягового электропривода, что позволяет достичь максимальной эффективности управления электроавтомобилем на повороте.

Общее СЭД на основе НС с линейной и нелинейной системами:

- V^* – общая скорость движения электроавтомобиля
- δ – угол поворота руля.
- Расчёт параметров $V1, V2$ – скорости вращения левого и правого задних колёс, соответственно.

Отличия СЭД на основе НС от линейной и нелинейной систем:

1. В учёте внутренних параметров электроавтомобиля.

2. В учёте внешних параметров электроавтомобиля.

3. Возможности интеграции системы в различные конфигурации компоновки тяговых электродвигателей, в технологии беспилотного наземного транспортного средства и вспомогательные системы управления автомобилем.

Преимущества СЭД на основе НС:

1. Подходит для любых компоновок привода автомобиля (2, 4, 6, 8 тяговых двигателей).

2. Учитывает конструктив и внутренние параметры автомобиля (колесная база, длина, радиус колёс, масса и др.).

4. Учитывает влияние погодных условий (температура, влажность, освещённость и др.);

5. Учитывает стиль вождения;

6. Пригодна для применения в беспилотном транспорте;

7. Отсутствует ограничение по диапазону скорости движения;

8. Модель НС легко заменить или обновить;

9. Учитывает состояние дорожного покрытия (асфальт, гравий, бездорожье);

10. Возможность интегрироваться в систему ABS, ESP, круиз-контроля и др;

Недостатки СЭД на основе НС:

1. Повышенные требования к вычислительным ресурсам.

Разработка СЭД на основе НС также имеет значение для беспилотного транспорта, т.к. позволяет повысить безопасность и эффективность движения. Благодаря использованию современных алгоритмов машинного обучения, интеграция разрабатываемой системы может осуществляться с минимальными затратами ресурсов и времени в действующих платформах беспилотного транспорта [30, 59, 85, 123, 127, 140].

На основе структурной схемы (см. рисунок 2.7) разработана функциональная схема системы электронного дифференциала, основанной на нейронной сети (НС), показанная на рисунке 2.8. Работа данной системы заключается в следующем. Микроконтроллер (МК) считывает показания с различных датчиков: скорости автомобиля (педаль акселератора), угла поворота рулевого колеса ($\delta_{руля}$),

гироскопа, акселерометра и датчика температуры. Данные с датчиков отправляются на персональный компьютер (ПК), где нейронная сеть обрабатывает информацию и формирует предсказание (задание) скорости вращения колёс. Значение данного задания возвращается на микроконтроллер, который управляет скоростью вращения тяговых двигателей (M1 и M2) посредством силовых преобразователей, что обеспечивает эффективную работу системы управления движением автомобиля при совершении поворотов и манёвров.

Математический аппарат работы нейронной сети для управления системой электронного дифференциала можно описать следующим образом. Нейронная сеть обучается на основе данных о динамике автомобиля для генерации выходных значений – скоростей вращения задних правого и левого колёс, что позволяет адаптивно управлять движением автомобиля во время поворотов и манёвров в зависимости от текущих условий. Основные этапы математического описания включают настройку входного и выходного слоёв, формирование функции активации и настройку функции потерь для оптимизации весов.

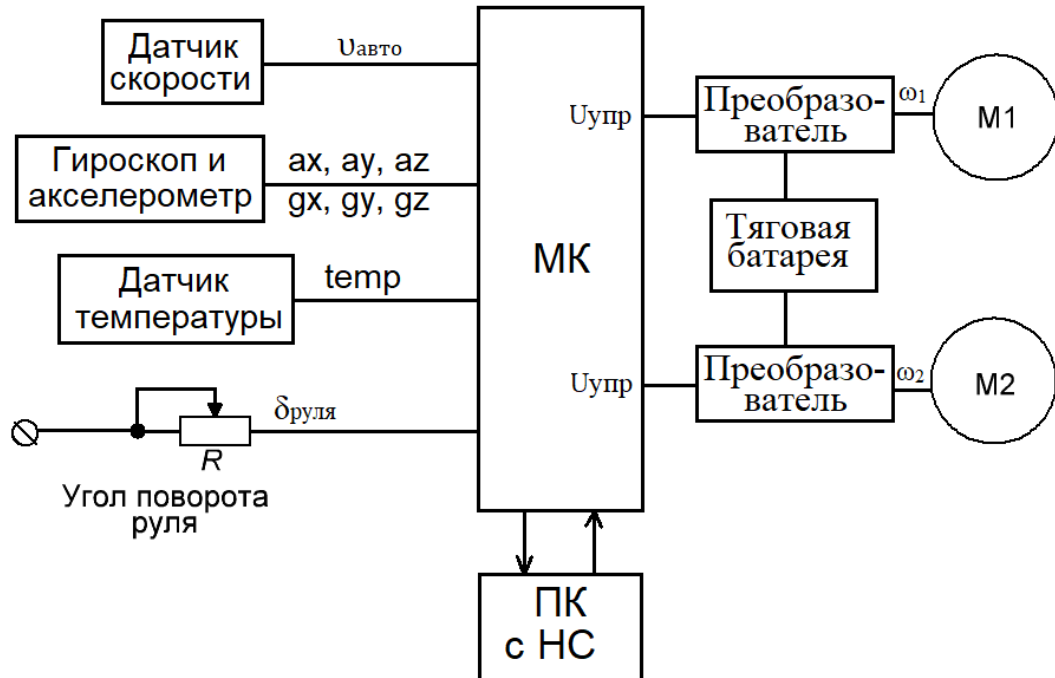


Рисунок 2.8 – Функциональная схема СЭД, основанной на НС

1. Входные данные с датчиков для НС

Обозначим входной вектор X следующим образом:

$$X = \begin{bmatrix} \omega_{\text{авто}} \\ a_x \\ a_y \\ a_z \\ g_x \\ g_y \\ g_z \\ temp \\ \delta_{\text{руля}} \end{bmatrix} \quad (2.7)$$

где $\omega_{\text{авто}}$ – скорость автомобиля;

a_x, a_y, a_z – угловое ускорение по оси «x», «y» и «z» соответственно (значения с датчика акселерометра);

g_x, g_y, g_z – ускорение свободного падения по оси «x», «y» и «z» соответственно (значения с датчика гироскопа);

$temp$ – температура окружающей среды;

$\delta_{\text{руля}}$ – угол поворота руля.

2. Архитектура нейронной сети

Нейронная сеть обычно включает несколько скрытых слоёв с нелинейными функциями активации. Обозначим структуру сети следующим образом:

- Входной слой: принимает вектор X
- Скрытые слои: содержат L скрытых слоёв, каждый с n_l нейронами на слое, где $l=1, \dots, L$.

- Выходной слой: генерирует выходной вектор Y с элементами ω_1, ω_2 , обозначающими скорости вращения задних левого и правого колёс, соответственно.

3. Вычисления внутри нейронной сети

На каждом скрытом слое l происходит линейное преобразование с добавлением смещения и активацией:

$$Z^l = W^l \cdot A^{l-1} + b^l, \quad (2.8)$$

$$A^l = \sigma(Z^l), \quad (2.9)$$

где Z^l – результат вычисления на слое l ;

W^l, b^l – матрица весов и вектор смещений на слое l ;

A^l – активация на слое l ;

$\sigma()$ – функция активации, например, ReLU или сигмоида, чтобы добавить нелинейность в модель.

Для конечного слоя выходные значения Y определяются следующим образом:

$$Y = W^{(L+1)} \cdot A^L \cdot b^{L+1}, \quad (2.10)$$

где W^{L+1} , b^{L+1} – матрица весов и вектор смещений для выходного слоя

4. Функция потерь и оптимизация

Целевая функция для обучения нейронной сети – это функция потерь, которая отражает разницу между предсказанными и реальными значениями скоростей колёс. Представим истинные значения скоростей вращения как:

$$Y_{true} = [\omega_{1true}, \omega_{2true}] \quad (2.11)$$

Тогда функция потерь, которая, например может быть выражена как среднеквадратическая ошибка (MSE) будет равна:

$$Loss = \frac{1}{2} \sum_i (Y_{pred}^i - Y_{true}^i)^2, \quad (2.12)$$

где Y_{pred}^i – i -е предсказание значения скорости вращения колёс от нейронной сети.

Для оптимизации весов W и смещений b используется метод обратного распространения ошибки в сочетании с алгоритмом градиентного спуска или его улучшенными версиями (например, Adam, AdaMax, RMSProp и др).

5. Выходные значения

После обучения сети её работа заключается в приёме новых данных о текущей скорости автомобиля, значениях с датчика акселерометра и гироскопа, температуре и угле поворота руля, на основе которых сеть предсказывает значения $Y=[\omega_1, \omega_2]$, что позволяет регулировать скорости вращения колёс и обеспечивать оптимальную управляемость автомобиля при манёврах и поворотах.

Этот подход позволяет учесть сложные зависимости между входными данными и необходимыми скоростями вращения колёс, адаптируясь к изменениям условий движения и улучшая управляемость автомобиля.

Выводы по главе 2

1. Инерционные силы, возникающие при движении автомобиля по криволинейной траектории, создают дополнительные нагрузки на элементы системы управления и сцепления шин с дорожным покрытием, что приводит к снижению эффективности управления электроавтомобилем.

2. Влияние параметров окружающей среды, таких как влажность, температура, обледенение и наличие осадков на дорожном покрытии, играют важную роль в поведении автомобиля при движении на поворотах. Изменение погодных условий существенно снижает коэффициент сцепления шин с дорогой, увеличивая риск заноса и снижая общую управляемость транспортного средства.

3. Использование системы электронного дифференциала, на основе нейронной сети позволяет решить проблему эффективного управления транспортом во время манёвров и поворотов на повышенных скоростях с учётом дополнительных факторов: центростремительное ускорение, параметры окружающей среды и стиль вождения.

ГЛАВА 3. РАЗРАБОТКА ФИЗИЧЕСКОЙ МОДЕЛИ ЭЛЕКТРОМОБИЛЯ И МЕТОДИКИ ПОЛУЧЕНИЯ ЭКСПЕРИМЕНТАЛЬНЫХ ДАННЫХ НЕОБХОДИМЫХ ДЛЯ ОБУЧЕНИЯ НС

Основной спецификой работы с алгоритмами машинного обучения, в частности, нейронными сетями, является наличие датасета – структурированного массива экспериментальных данных, которые будут использоваться в процессе обучения. Для этих целей требуется физическая модель электромобиля, которая должна отвечать следующим техническим требованиям:

1. Механическая часть.

-Шасси и корпус. Конструкция должна быть достаточно прочной и лёгкой, чтобы выдерживать нагрузки от движущихся компонентов и внешние воздействия, сохраняя при этом устойчивость и манёвренность.

-Подвеска. Подвеска должна обеспечивать устойчивость и плавность хода, а также быть адаптирована для работы с задними тяговыми двигателями.

-Рулевая система. Конструкция рулевой системы должна обеспечивать управляемость модели с учётом поворота колёс и обеспечивать повороты с разными радиусами.

-Тяговые двигатели. Задняя ось оснащается двумя двигателями для независимого управления скоростью левого и правого колеса. Мощность двигателей должна быть достаточной для преодоления сопротивлений движения и инерционных нагрузок.

-Система передачи вращения. Для передачи вращения от двигателей должны использоваться зубчатые ремни и шестерни. Ремни должны быть износостойкими, с точной передачей момента, а шестерни – спроектированы с минимальным люфтом для устойчивого движения.

2. Электронная система управления.

-Микроконтроллер. Система управления должна быть построена на базе микроконтроллера с достаточной вычислительной мощностью для обработки сигналов от датчиков и управления двигателями. Микроконтроллер должен

поддерживать возможность обновления прошивки для внесения изменений и оптимизации алгоритмов управления.

-Интерфейс связи с ПК. Потребуется наличие интерфейса для обмена данными с ПК, например, USB или Bluetooth, для передачи данных в реальном времени и настройки параметров управления.

3. Система датчиков.

-Гироскоп и акселерометр. Система датчиков должна обеспечивать измерение угловой скорости, продольных и поперечных ускорений для определения параметров движения и поворота.

-Датчики скорости колес. Необходимо применение датчиков для измерения скорости вращения каждого заднего колеса для реализации системы электронного дифференциала.

-Датчик угла поворота. Необходим датчик угла поворота рулевого механизма для обеспечения точной информации о положении передних колёс.

4. Силовая часть и компоненты управления двигателями.

-Силовые транзисторы. Для управления двигателями должны использоваться силовые элементы, рассчитанные на соответствующие токовые и тепловые нагрузки, обеспечивающие быструю реакцию и стабильность работы.

-Печатная плата. Все компоненты силовой и управляющей части должны быть расположены на печатной плате с оптимальным расположением трассировки, минимальным уровнем электромагнитных помех и достаточным охлаждением для работы при различных нагрузках.

5. Программное обеспечение и алгоритмы управления.

-Разработанный алгоритм должен позволять управлять скоростью и направлением движения каждого колеса посредством микроконтроллера и ПК.

-Интерфейс калибровки. Необходима возможность калибровки датчиков и настроек управления через ПК для точного воспроизведения модели поведения автомобиля в различных условиях.

3.1. Разработка физической модели электромобиля

Этапы разработки и изготовления технической части физической модели электромобиля:

1. Разработка функциональной схемы электропривода электромобиля и системы управления;
2. Выбор компонентной базы для тягового электропривода, системы управления и датчиков;
3. Разработка принципиальной схемы и печатной платы тягового электропривода и системы управления;
4. Разработка передаточного устройства для тяговых двигателей;
5. Разработка управляющего программного обеспечения (ПО) для модели электромобиля.

Разработка функциональной схемы электропривода электромобиля и системы управления

На рисунке 3.1 представлена функциональная схема управления моделью электромобиля с использованием системы электронного дифференциала на основе нейронной сети. Основными элементами системы являются:

1. Тяговая батарея – обеспечивает питание всей системы, в том числе через преобразователь постоянного тока (DC/DC), который регулирует напряжение для различных вспомогательных компонентов.

2. Микроконтроллер – центральный управляющий элемент системы, который принимает сигналы от сенсоров и передает управляющие команды. Он получает данные с:

-Гироскопа и акселерометра для определения ускорения и угловой скорости вращения вокруг оси (g_x , g_y , g_z , a_x , a_y , a_z) модели электроавтомобиля.

-Датчика температуры для измерения температуры окружающей среды.

-ПК с нейронной сетью (НС) – через интерфейс UART для получения и обработки данных с нейросетевого алгоритма управления.

3. Датчики скорости, установленные на каждом из двух электродвигателей (M1 и M2), измеряют текущую скорость каждого колеса и передают информацию на микроконтроллер.

4. Драйверы для двигателей – представлены в виде H-моста и нужны для управления работой каждого электродвигателя (M1 и M2) на основе команд, поступающих от микроконтроллера. Управление осуществляется с помощью широтно-импульсной модуляции (ШИМ).

5. Сервопривод рулевого управления – отвечает за поворот рулевого механизма, управляется через сигнал угла руля, который поступает от микроконтроллера.

Весь процесс управления замкнут в контуре обратной связи, где микроконтроллер анализирует данные с сенсоров и ПК с нейронной сетью, рассчитывает необходимые корректировки для поддержания оптимального движения и передает команды на драйверы двигателей и сервопривод рулевого управления, что обеспечивает устойчивость и эффективность электромобиля при повороте или манёвре.

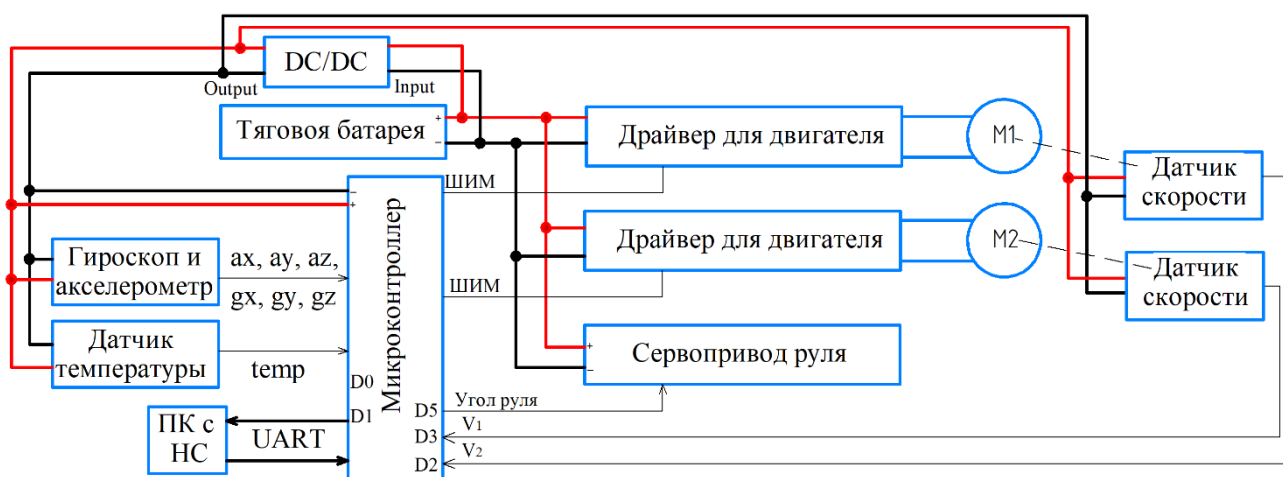


Рисунок 3.1 – Функциональная схема физической модели электромобиля

Для измерения ускорения и положения в пространстве (необходимость этого процесса указана в главе 2.1) используется специализированный датчик «MPU6050» в виде готового модуля для удобного подключения к микроконтроллеру по интерфейсу I²C. Гироскоп измеряет угловую скорость вращения вокруг оси, в градусах/секунду и ускорение в «g» (рисунок 3.2).

MPU-6050 представляет собой интегрированное 6-осевое устройство для отслеживания движения, объединяющее в себе 3-осевой гироскоп и 3-осевой акселерометр, дополненные встроенным цифровым процессором движения (Digital Motion Processor, DMP). Это устройство обладает возможностью интеграции с внешними неинерциальными цифровыми датчиками, например, датчиками давления, через вспомогательный I²C-порт, что расширяет возможности для создания сложных систем контроля движения.

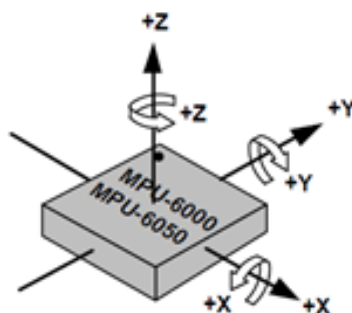


Рисунок 3.2 – Полярность осей чувствительности датчика MPU-6050.

Для оцифровки данных MPU-6050 оснащен тремя 16-битными аналого-цифровыми преобразователями (АЦП), которые используются для обработки выходных сигналов гироскопа, и тремя 16-битными АЦП для оцифровки данных акселерометра. Диапазоны измерения гироскопа могут быть установлены пользователем и варьируются от ± 250 до $\pm 2000^\circ/\text{сек}$ (dps), что позволяет адаптировать устройство под различные условия измерения угловых скоростей. Акселерометр также программируемый, с возможностью выбора диапазонов измерений: ± 2 g, ± 4 g, ± 8 g и ± 16 g, что делает его универсальным для применения как в системах с высокими требованиями к точности, так и в устройствах с большими динамическими нагрузками.

При помощи трёхосевого гироскопа и трёхосевого акселерометра, установленных на автомобиль, можно измерить и понять следующие параметры:

1. Угловые скорости. Гироскоп измеряет угловые скорости вращения автомобиля вокруг трёх осей (оси X, Y, Z). Это позволяет определить:

-*Тангаж* (вращение вокруг поперечной оси, например, при подъёме или спуске автомобиля).

-*Крен* (вращение вокруг продольной оси, например, при поворотах).

-*Рыскание* (вращение вокруг вертикальной оси, например, при изменении направления движения).

2. Ускорения по осям. Акселерометр фиксирует линейные ускорения по трём осям, что помогает измерить:

-*Продольное ускорение* (вдоль оси X), которое может возникать при разгоне или торможении автомобиля.

-*Поперечное ускорение* (вдоль оси Y), важное при анализе боковых сил, например, при поворотах.

-*Вертикальное ускорение* (вдоль оси Z), которое связано с движением по неровностям или при прохождении подъёмов и спусков.

3. Положение в пространстве. На основе данных акселерометра можно определить текущее положение автомобиля относительно вертикали, а также следить за отклонениями, например, при крене автомобиля или подъёме/спуске.

4. Силы, действующие на автомобиль. По данным акселерометра можно оценить силы инерции, действующие на автомобиль при различных манёврах, что полезно для анализа устойчивости, управления и комфорта вождения.

5. Динамику автомобиля. Совместный анализ данных гироскопа и акселерометра позволяет получить полную картину динамики движения автомобиля, включая его траекторию, характер движения на поворотах, реакции на неровности дороги и влияние ускорений на элементы автомобиля.

6. Определение заноса автомобиля. Трёхосевой гироскоп и акселерометр могут помочь в определении заноса автомобиля. Занос сопровождается резким изменением угловой скорости вокруг вертикальной оси (рыскание), что фиксируется гироскопом. Одновременно акселерометр покажет аномальное поперечное ускорение, характерное для скольжения автомобиля боком. Если угол заноса превышает определённое значение, это может свидетельствовать о потере сцепления с дорогой и переходе в скольжение.

7. Определение проскальзывания ведущих колёс. Проскальзывание ведущих колёс можно определить на основе несоответствия между ускорениями,

измеренными акселерометром, и угловыми скоростями колёс, полученными от других сенсоров (например, датчиков скорости). При проскальзывании угловая скорость ведущих колёс будет расти, но фактическое линейное ускорение автомобиля, измеряемое акселерометром, не будет соответствовать этому приросту.

8. Определение типа покрытия (асфальт или грунт). Анализ динамики ускорений и угловых скоростей может также помочь в определении типа дорожного покрытия. При движении по асфальту показатели будут более стабильны, так как сцепление с дорогой выше. На грунте, напротив, можно ожидать резких колебаний в показаниях акселерометра (особенно по вертикальной оси), что связано с неровностями покрытия, а также изменения угловых скоростей из-за возможного проскальзывания и меньшего сцепления.

Выбор компонентной базы, разработка принципиальной схемы тягового электропривода с системой управления и печатной платы.

На основе функциональной схемы (рисунок 3.1) была разработана принципиальная схема модели электромобиля с указанием перечня используемых элементов (*Приложение А*). На печатной плате размещены все основные компоненты системы электронного дифференциала, включая управляющую часть, систему датчиков, силовую часть, а также элементы индикации и защиты, представленные в *Приложении Б*. Как уже отмечалось ранее, модель предназначена для получения данных для нейронной сети и проверки концепции работоспособности системы электронного дифференциала для электрического автомобиля, основанной на НС, в целом.

Принципиальная схема содержит в себе силовую часть электропривода, систему управления, систему датчиков, передаточных устройств и вспомогательных элементов. В качестве тягового привода использовались двигатели постоянного тока из-за простоты монтажа и управления, Н-мост в качестве силовой части питания электродвигателя, литиевые аккумуляторы LiitoKala HG2 в качестве накопителей энергии. Для Н-моста, представленным MOSFET-транзисторами был разработан драйвер для управления затворами на базе

биполярных NPN и PNP-транзисторов с гальванической развязкой в виде оптопары PC817. В качестве модуля связи с ПК использовался bluetooth модуль HC-06, который подключался к UART-интерфейсу микроконтроллера для передачи и приёма данных с управляющего всей системой компьютера.

Разработка физической модели электроавтомобиля и передаточного устройства.

Из базовой платформы для гоночной радиоуправляемой модели автомобиля (см. рисунок 3.3) были удалены оригинальные компоненты силовой части электропривода и блок управления. При этом сохранены элементы, такие как привод рулевых тяг, подвеска и верхняя часть корпуса. Замена старых компонентов позволила значительно усовершенствовать модель для дальнейшего использования в экспериментальных целях. На место удалённых компонентов была установлена новая аккумуляторная батарея и обновлённые тяговые приводы, которые позволят проводить эксперименты работы СЭД.

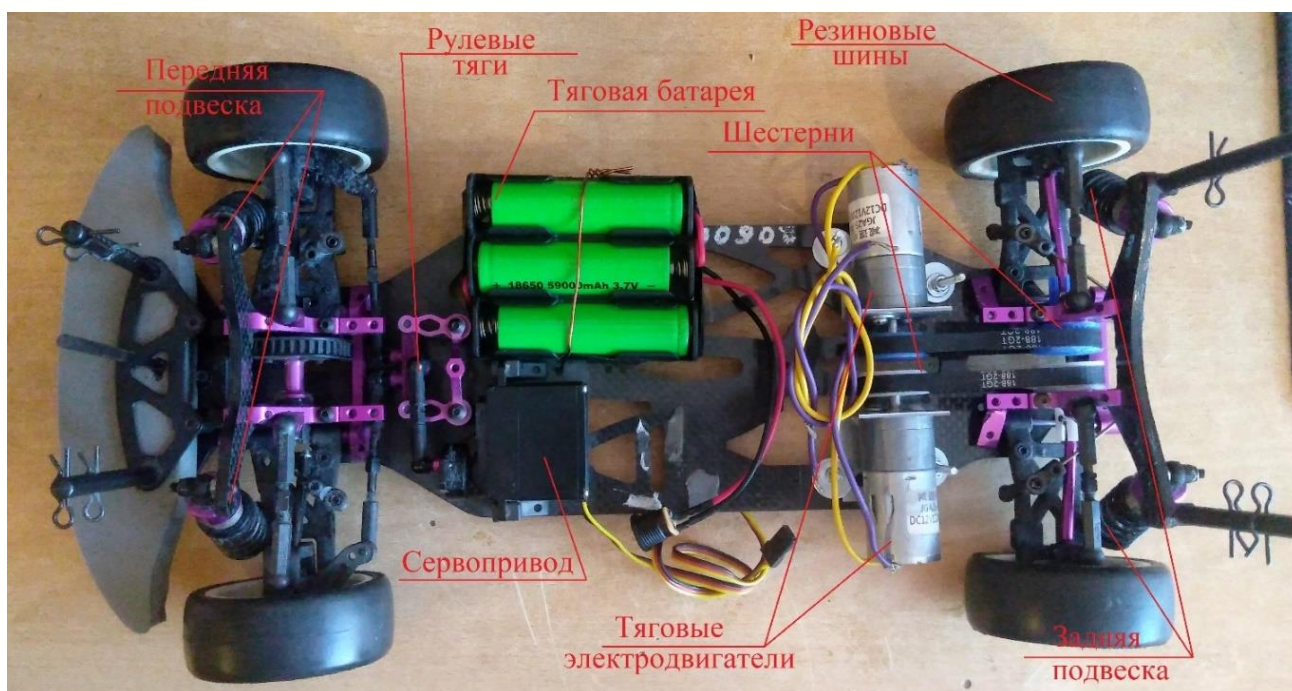


Рисунок 3.3 – Платформа для разработки модели электроавтомобиля

Для передачи вращения от электродвигателей к колёсам автомобиля использовался зубчатый ремень – «GT2 Timing Belt» с шагом 2 мм, что позволило минимизировать потери при передаче энергии, как это было бы в случае с гладким ремнём. Проектирование шестерён, как ведомых, так и ведущих (см. рисунок 3.3),

под шаг ремня осуществлялось в программном комплексе «Fusion 360». Для создания шестерён использовался плагин «Multi Spur Gear», который позволил точно рассчитать и смоделировать параметры зубьев (см. Приложение В). Такой подход к проектированию и замене компонентов существенно улучшил характеристики модели и её пригодность для проведения дальнейших исследований и экспериментов.

Разработка управляющего программного обеспечения (ПО) для модели электромобиля, для получения экспериментальных данных.

Основной алгоритм управления моделью электроавтомобиля и получения экспериментальных данных для обучения электронной дифференциальной системы базируется на обработке пакетов данных, поступающих от микроконтроллера и направляемых к нему. Эти пакеты содержат информацию о динамических параметрах автомобиля, таких как скорость колес, угол поворота и др.

На первом этапе получения данных для датасета, управление происходит в полуручном режиме – здесь оператор задает текущий угол поворота колёс при помощи сервопривода, скорость приводов (значение от 0 до 255 – размерность ШИМ ARV-контроллера) и разрешение на движение (в реальном времени). Собранные данные с датчиков, а также вручную занесенное значение отклонения от идеальной траектории в таблицу в конечном итоге будут масштабированы в диапазон значений $[-1;1]$, это требуется для лучшего схождения градиентов и соответственно улучшения точности предсказания нейронной сети.

Управление исполнительными устройствами происходит посредством микроконтроллера. Самые сложные процессы, такие, как например, работа нейронной сети, считывание управляющих значений оператора, обработка и фильтрация данных и т.п. происходят посредством компьютера. Для микроконтроллера отправляется лишь пакет данных на исполнение и пакет данных с датчиков. Дефолтный (по умолчанию) пустой пакет данных (данные типа «строка») для управления может быть представлен в следующем виде:

"00000000000000",

где 0-е число означает старт работы и может принимать значение «0», либо «1» и соответственно запрещает или разрешает работу МК;

1-е число обозначает состояние на отправку данных микроконтроллером может принимать значение «0», либо «1» и соответственно запрещает или отправку данных МК;

2-е число – направление движения, может принимать значение «0», либо «1» то есть движение вперёд или назад;

3-е число – разрешение на ускорение, может принимать значение «0», либо «1» то есть сигнал на ускорение или его отсутствие;

числа с 4 по 6 задают скорость левого двигателя (от 0 до 255);

числа с 7 по 9 задают скорость правого двигателя (от 0 до 255);

числа с 10 по 12 задают угол сервопривода (от 25 до 155 градусов, данные значения получены, исходя из возможностей модели);

Измеренные микроконтроллером экспериментальные данные отправляются на ПК посредством кода, с уточняющими комментариями, показанного ниже:

```
void SendData() {
  mpu.getMotion6(&ax, &ay, &az, &gx, &gy, &gz); // Получение данных с
  датчика гироскопа и акселерометра
  temp = mpu.getTemperature(); // Получение значения температуры воздуха
  Serial.print(LeftMotorSpeed); Serial.print("\t"); // Отправка текущей
  скорости левого заднего двигателя
  Serial.print(RightMotorSpeed); Serial.print("\t");// Отправка текущей
  скорости правого заднего двигателя
  Serial.print(ServoAngle); Serial.print("\t"); //Отправка угла поворота
  руля
  Serial.print(ax); Serial.print("\t"); //Отправка углового ускорения по
  оси X
  Serial.print(ay); Serial.print("\t"); // Отправка углового ускорения по
  оси Y
  Serial.print(az); Serial.print("\t"); // Отправка углового ускорения по
  оси Z
  Serial.print(gx); Serial.print("\t"); // Отправка значения гироскопа по
  оси X
  Serial.print(gy); Serial.print("\t"); // Отправка значения гироскопа по
  оси Y
  Serial.print(gz); Serial.print("\t"); // Отправка значения гироскопа по
  оси Z
  Serial.print(temp); Serial.print("\t"); // Отправка значения температуры
  воздуха
  Serial.println(); // Каретка конца пакета
}
```


После принятия компьютером этих данных, они структурируются в массив и экспортируются в «excel»-файл для последующей обработки в датасет, пригодный для обучения нейронной сети электронного дифференциала для электромобиля, в соответствии с блоком кода, с комментариями, показанного ниже:

```
# Функция для экспорта необработанных данных в excel
def ExportData():
    data_dict['V1'] = dataLists[0] #Скорость левого заднего двигателя
    data_dict['V2'] = dataLists[1] #Скорость правого заднего двигателя
    data_dict['AngleServo'] = dataLists[2] #Угол поворота руля
    data_dict['ax'] = dataLists[3] #Угловое ускорение по оси X
    data_dict['ay'] = dataLists[4] #Угловое ускорение по оси Y
    data_dict['az'] = dataLists[5] #Угловое ускорение по оси Z
    data_dict['gx'] = dataLists[6] #Значение гироскопа по оси X
    data_dict['gy'] = dataLists[7] #Значение гироскопа по оси Y
    data_dict['gz'] = dataLists[8] #Значение гироскопа по оси Z
    data_dict['temp'] = dataLists[9] #Значение температуры

    write_to_excel(excel_filename, data_dict) # Обновим данные в
    excel-файле
    result_array.clear() # Очистим массив от старых значений
```

3.2 Методика получения экспериментальных данных для обучения НС

Этапы разработки методики получения экспериментальных данных для обучения НС и их обработки:

1. Разработка методики для получения экспериментальных данных, для обучения НС;
2. Анализ рекомендаций при создании релевантного набора обучающих данных, а также разработка ПО для получения и обработки данных;
3. Выбор подходящего типа нейронной сети для реализации системы электронного дифференциала;
4. Разработка ПО для обучения нейронной сети на основе полученных данных.

3.2.1. Методика получения экспериментальных данных

Т.к. ранее не существовало установок подобного типа, прежде, чем начать процесс получения данных необходимо разработать способ качественного обучения НС. Способ должен позволить нейронной сети иметь как положительный пример для обучения, так отрицательный и нейтральный, что позволит создать более устойчивую систему к любым внешним факторам. Также методика обучения

должна легко масштабироваться для реализации расширенных систем с возможностью внедрения в иные электротехнические комплексы.

В данной работе предложена методика обучения нейронной сети на основе карт поворотов, учитывающая величину отклонения от идеальной траектории и показания установленных на модели электромобиля датчиков в этот момент. Предложенный метод легко масштабируется, поддается автоматизации и не требует коррекций при изменении количества измеряемых параметров. Важно отметить, что идеальная траектория задается разработчиком и может основываться как на реальных переходных кривых, применяемых при проектировании дорог общего пользования, так и на специфических траекториях, характерных для гоночного или специализированного транспорта.

Ниже показано несколько подходов к построению идеальных траекторий поворота, в зависимости от условий и характера движения.

1. Идеальные траектории на основе чистых кривых при свободном движении колес.

Одним из вариантов построения идеальных траекторий является использование так называемых "чистых" кривых, которые формируются при движении модели с полностью свободным вращением всех колес и фиксированным углом поворота руля, которой был использован в данной исследовательской работе. В этом случае траектория получается естественной, основанной на физических характеристиках транспортного средства и его взаимодействии с поверхностью. Визуализация использованной карты поворотов, составленной на миллиметровой бумаге показана на рисунке 3.4. Применение масштабной координатной сетки упрощает вычисления отклонений от идеальной траектории, что удобно при последующей обработке данных.

2. Использование переходных кривых.

Для более универсальных условий, например, при движении по дорогам, в качестве идеальных траекторий могут применяться переходные кривые, которые используются при проектировании дорог общего пользования. Переходные кривые учитывают требования безопасности и плавности движения, что особенно важно

при создании дорог для общественного транспорта. Такие кривые обеспечивают комфортное и безопасное прохождение поворотов на средних скоростях и могут быть использованы в качестве эталона для тренировки нейронной сети.

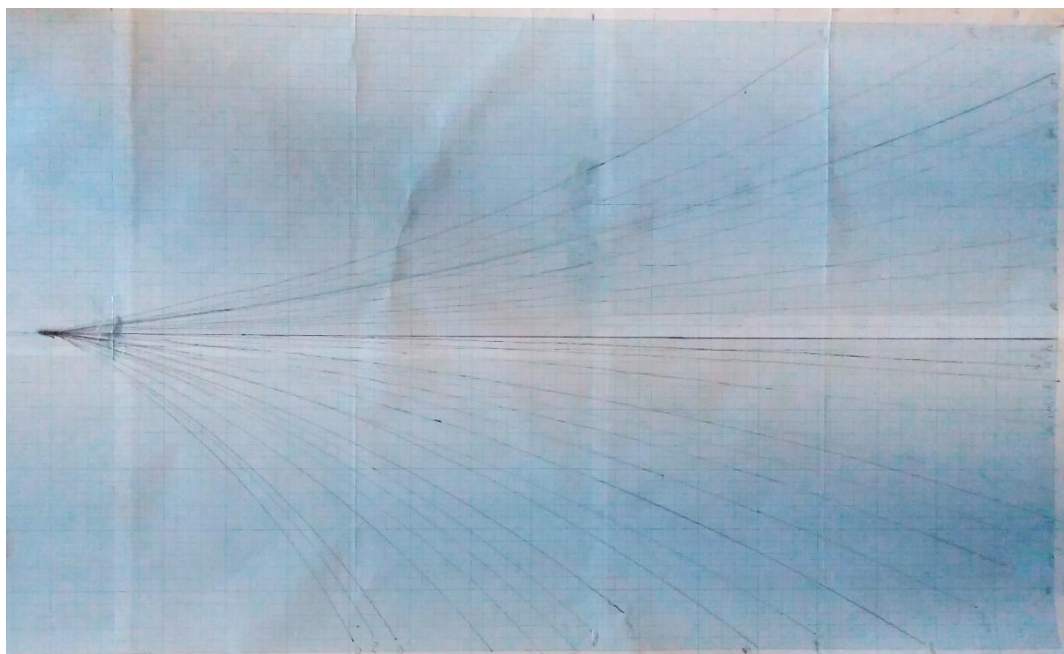


Рисунок 3.4 – Использованная карта идеальных траекторий для получения данных для обучения НС

Пример построения таких кривых показан на рисунке 3.5. Сокращения на нём имеют следующие определения: НПК / КПК – начало / конец поворота кривой; НКК / ККК – начало / конец круговой кривой, геометрические элементы закругления: угол – α , радиус – R , кривую – K , тангенс – T , биссектрису – B , домер – T_0 и длину переходных кривых L . Домер используется для обозначения части переходной кривой, которая находится между прямым участком дороги и кривой. Методика расчёта таких кривых регламентируется нормативным документом «СП 34.13330.2021. Свод правил. Автомобильные дороги. СНиП 2.05.02-85».

3. Оптимальные траектории для гоночных условий.

Для специфичных задач, таких как прохождение поворотов на гоночных трассах, важны оптимальные траектории, которые позволяют максимально эффективно использовать сцепление шин с дорогой и минимизировать потерю скорости. В условиях высокой скорости идеальная траектория прохождения поворота может быть рассчитана с учетом гоночных стратегий, включая

определяют, как транспортное средство будет смещаться боком относительно направления движения.

Карты заноса могут быть разработаны на основе данных о трении, угле заноса и других параметров, специфичных для дрифта, как показано оранжевой линией на рисунке 3.7, на рисунке:

1. Угол заноса. Оценивается по углу автомобиля относительно идеальной гоночной траектории для автомобилей, которые нормально сцепляются с трассой (выше лучше).

2. Расстояние от гоночной траектории для автомобилей, которые нормально сцепляются с трассой (меньше лучше).

3. Скорость движения (выше лучше).

4. Продолжительность заноса (дольше лучше).

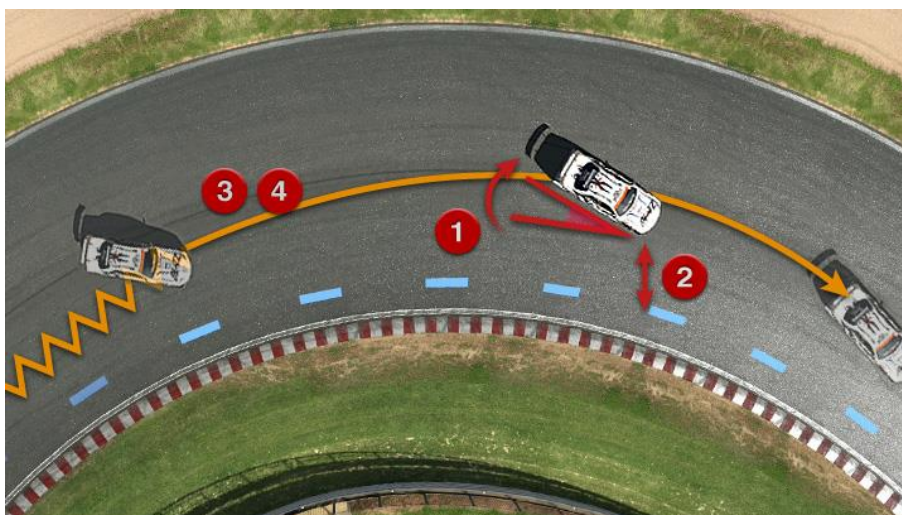


Рисунок 3.7 – Пример идеальной траектории прохождения дрифт-трассы

5. Траектории для движения по неустойчивым поверхностям.

В условиях движения по нестабильным поверхностям, таким как песок, грязь или снег, также необходимо учитывать специфические траектории. Идеальные траектории для этих условий разрабатываются с учетом поведения транспортного средства на таких поверхностях, где возможны пробуксовки, сниженное сцепление и увеличение времени реакции. Для корректной оценки траекторий используются данные о взаимодействии колес с деформируемыми поверхностями, что позволяет обучить нейронную сеть учитывать особенности таких покрытий. Данная вариация

разработанной методики может быть применена, например, в раллиных гонках и для внедорожного транспорта.

6. Траектории для автономного транспорта.

В условиях развития автономного вождения одним из ключевых факторов является оптимизация траекторий для безопасного и плавного движения без участия человека. В данном случае идеальная траектория может основываться на сложных алгоритмах прогнозирования, которые учитывают дорожные условия в реальном времени, наличие других транспортных средств, пешеходов и динамических препятствий. Эти алгоритмы могут включать элементы машинного зрения, распознавания объектов и прогнозирования их поведения.

7. Траектории для городской среды с переменными ограничениями.

В условиях плотного городского движения и переменных ограничений, таких как знаки, светофоры, пешеходные переходы и зоны парковки, идеальная траектория должна учитывать динамику дорожного потока и изменяющиеся условия. Это может включать в себя предсказание оптимальной полосы движения, минимизацию задержек на светофорах, а также учет внезапных изменений дорожных ситуаций (например, появление препятствия). В таких условиях может быть полезным использование карт, которые адаптируются к времени суток или различным сценариям движения (час пик, ночное время).

8. Траектории для транспортных средств с прицепом.

Особым случаем могут быть траектории для транспортных средств с прицепом (грузовики, фуры и автопоезда). В этих случаях идеальная траектория должна учитывать не только маневренность ведущего автомобиля, но и поведение прицепа, который следует за ним с определенным запозданием и по иной траектории (рисунок 3.8). Это особенно важно при прохождении поворотов или выполнении парковки в ограниченном пространстве.

9. Траектории для транспорта с изменяемой геометрией.

Для транспортных средств с изменяемой геометрией, таких как грузовые платформы с регулируемой шириной осей или длиной шасси, идеальная траектория может зависеть от текущей конфигурации транспорта. В зависимости

от задачи, например максимальная маневренность или устойчивость при высоких скоростях, траектория движения может динамически изменяться в зависимости от изменения параметров транспортного средства в реальном времени. Это позволяет адаптировать траекторию для эффективного преодоления различных типов дорог и препятствий.

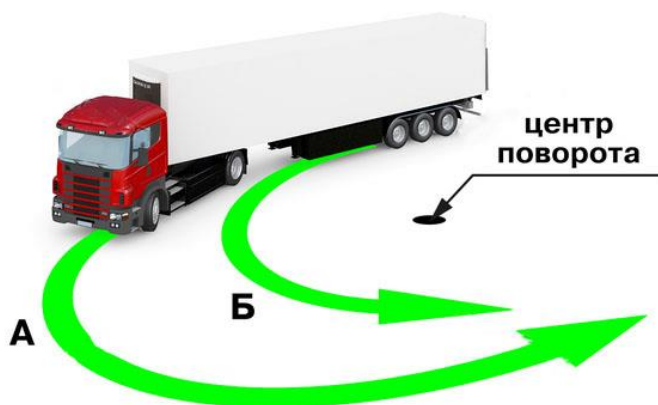


Рисунок 3.8 – Траектория движения транспорта с прицепом

Таким образом, выбор идеальной траектории определяется конкретными условиями эксплуатации транспортного средства, а также характером дорожного покрытия и задачами, стоящими перед системой управления.

Способ вычисления отклонения при повороте модели от идеальной траектории для обучения нейронной сети электронного дифференциала показан на рисунке 3.9. Принцип состоит в том, чтобы заранее расчертить на карте поворотов идеальные траектории первого типа – «траектории на основе чистых кривых» (красная линия на рисунке 3.9), к которым нужно стремиться при управлении приводами при помощи НС. При проведении экспериментов на тяговые приводы подаются различные вариации скорости вращения, чтобы получить иную траекторию движения (синяя линия на рисунке 3.9), данные с датчиков и самое главное величину отклонения от идеальной траектории в градусах. Именно эти данные после фильтрации и структуризации будут использованы для обучения НС. При штатной работе системы электронного дифференциала данное отклонение будет автоматически задаваться нулём, что позволит электромобилю всегда следовать идеальной траектории вне зависимости от внешних условий.

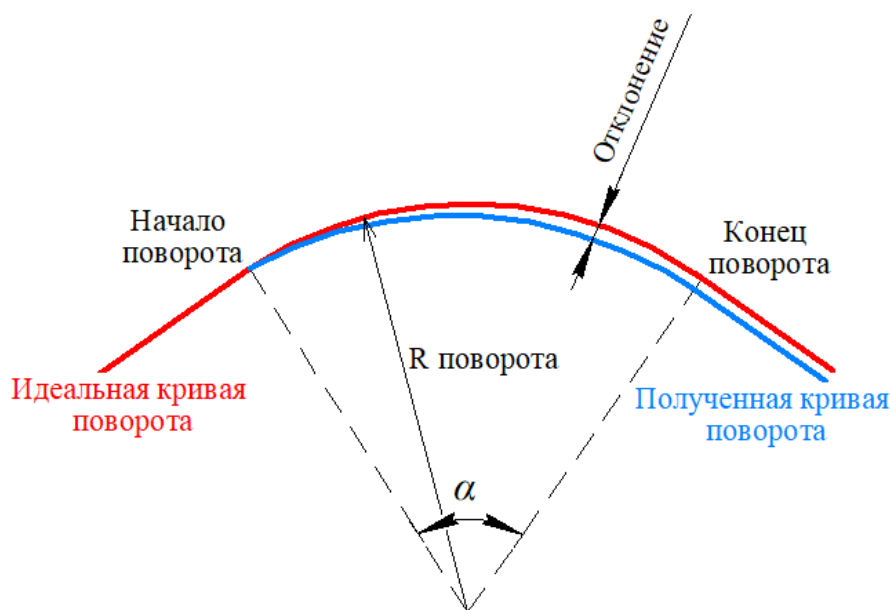


Рисунок 3.9 – Принцип вычисления отклонения модели от идеальной траектории поворота для обучения НС

Данные, полученные в ходе экспериментов, после обработки будут использованы для обучения нейронной сети системы электронного дифференциала. Структура этого процесса представлена на рисунке 3.10. На нем показаны значения скорости вращения колёс (ω_1, ω_2), задающее значение скорости движения транспортного средства ($v_{\text{авто}}$), показания датчика гироскопа (g_x, g_y, g_z), датчика акселерометра (a_x, a_y, a_z), датчика температуры ($temp$), измеренное значение отклонения от идеальной траектории ($deviation$) и угол поворота руля для сервопривода ($\delta_{\text{руля}}$).

3.2.2. Методика обработки данных для создания датасета

Данные являются важным компонентом любой модели искусственного интеллекта, достаточные объемы данных позволяют анализировать тенденции и скрытые закономерности и принимать решения на основе построенного датасета.

Согласно рекомендациям разработчиков из Google при составлении датасета следует следовать 4 следующим шагам:

1. Собрать исходные данные.
2. Определить независимые данные и сделать их маркировку.
3. Разделить данные тренировочные и тестовые.

4. Смасштабировать данные.

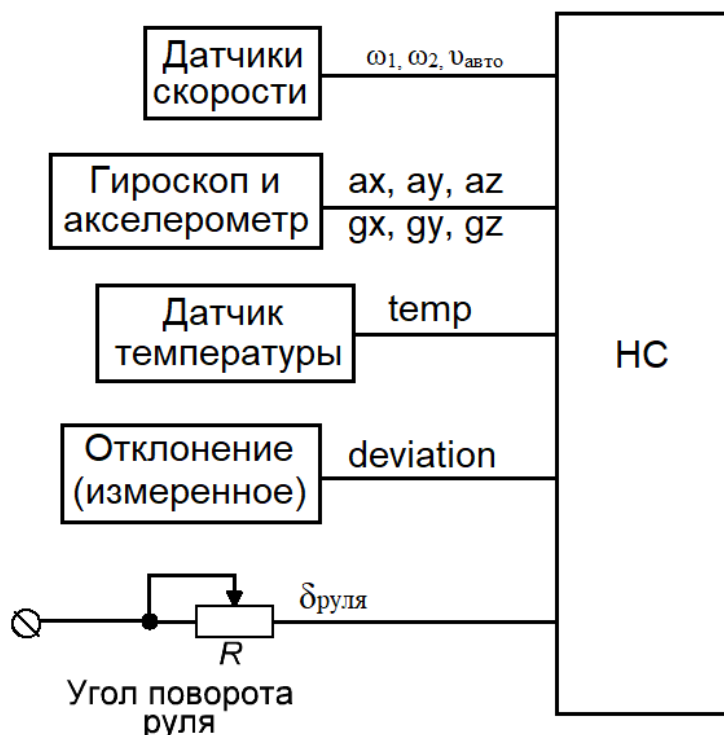


Рисунок 3.10 – Структурная схема использования экспериментальных данных для обучения НС СЭД

Что касается размера набора данных, есть грубое эмпирическое правило: модель должна обучаться количеством примеров как минимум на порядок больше, чем число независимых параметров. Простые модели на больших наборах данных обычно превосходят сложные модели на небольших наборах данных. Примеры количества данных для нескольких датасетов представлены в таблице 3.1.

Таблица 3.1 – Пример размера датасета

Название датасета	Размер (количество примеров)
Iris flower data set	150 (всего)
MovieLens (the 20M data set)	20,000,263 (всего)
Google Gmail SmartReply	238,000,000 (тренировочный набор)
Google Books Ngram	468,000,000,000 (всего)
Google Translate	триллионы

Размер данных важен и может существенно влиять на процесс обучения модели и её производительность. Однако оптимальный размер данных зависит от

многих факторов, включая сложность задачи, архитектуру модели, количество параметров модели и доступные вычислительные ресурсы.

Вот несколько рекомендаций относительно данных:

1. *Маленький объем данных.* Маленький датасет может привести к переобучению, особенно если модель сложная. В таких случаях важна регуляризация модели или использование предобученных моделей и трансферного обучения.

2. *Средний объем данных.* Средний объем данных в большинстве случаев может быть достаточным для обучения более сложных моделей. Однако всегда стоит стремиться к разнообразию исходных данных, чтобы модель была способна лучше обобщать.

3. *Большой объем данных.* Большие датасеты обычно способствуют лучшей обобщающей способности модели. Однако в этом случае может потребоваться больше вычислительных ресурсов для обучения таких моделей.

4. *Гиперпараметры модели.* В случае, когда имеется ограниченный объем данных, нужно выбирать более простые модели или использовать методы регуляризации, чтобы избежать переобучения.

5. *Кросс-валидация.* Важно использовать кросс-валидацию при оценке производительности модели. Это помогает убедиться, что модель обобщается хорошо и не зависит от конкретного разбиения данных.

6. *Увеличение данных (Data Augmentation).* При ограниченном объеме данных, можно рассмотреть техники увеличения данных, которые позволяют создавать новые обучающие примеры из существующих.

Важно заметить, что баланс между размером данных и сложностью модели может быть достигнут только опытным путем. Нужно регулировать параметры модели и размер датасета в зависимости от результатов на валидационных данных, чтобы достичь оптимального компромисса между предсказательной силой и ресурсами для обучения.

Также для создания качественного датасета требуется выполнить следующие пункты:

1. *Определить цель и задачи.* Нужно четко определить какие цели требуется достичь при использовании датасета, например, для задачи классификации, регрессии, детекции объектов и т. д.

2. *Выбор источников данных.* Нужно определить источник, откуда будут получены данные. Это могут быть открытые наборы данных, базы данных, API или же собственные данные.

3. *Соблюдение разнообразия.* Нужно стремиться к разнообразию данных в применяемом датасете, это поможет модели лучше обобщать и обрабатывать различные сценарии.

4. *Очистка данных.* Внимательно просмотреть данные и устранить возможные ошибки, выбросы, дубликаты или отсутствующие значения, что позволит получить более точные и надежные результаты.

5. *Нормализация и стандартизация.* Привести все данные к одному стандарту. Например, нормализация числовых значений для того, чтобы они находились в одном диапазоне.

6. *Разделение на обучающую и тестовую выборки.* Разделить датасет на две части: обучающую выборку, которую модель будет использовать для обучения, и тестовую выборку, которую будет использована для проверки точности прогнозирования модели.

7. *Обработка категориальных переменных.* Если данные содержат категориальные переменные (имена, названия, обозначения, метки), нужно преобразовать их в числовой формат (0,1,2 и т.д.), используя кодирование или векторизацию.

8. *Балансировка классов* (при необходимости). Если классы данных несбалансированы, рассмотреть методы балансировки данных, чтобы модель не смещалась в сторону преобладающего класса.

В данной работе датасет будет иметь вид csv-таблицы со значениями, полученными с датчиков. Датасет данного типа можно будет создать 2-мя способами: при помощи библиотеки «pandas», либо заполнением вручную таблицы excel. При создании датасета «на ходу» можно использовать следующий пример:

```

import pandas as pd

# экспериментальные данные
data = {
    'Имя': ['Эксперимент 1', 'Эксперимент 2', 'Эксперимент 3'],
    'Параметр1': [10, 15, 20],
    'Параметр2': [0.5, 0.7, 0.3],
    'Результат': [100, 150, 200]
}

# Создание DataFrame из данных
df = pd.DataFrame(data)

# Сохранение DataFrame в CSV-файл
df.to_csv('экспериментальные_данные.csv', index=False)

```

В работе было принято решение использовать комбинированный метод – данные с датчиков при помощи библиотеки «pandas» сохраняются в excel-файле, а после происходит их ручная предобработка (фильтрация, форматирование, масштабирование, маркировка и прочее).

Итак, создание CSV-файла из экспериментальных данных в Excel может быть выполнено следующим образом:

1. Открыть Excel и ввести данные:

- Открыть новый лист Excel.
- Ввести экспериментальные данные, размещая их в различных столбцах и строках.

2. Сохранить файл как CSV:

- После ввода данных, выбрать "Файл" в верхнем левом углу.
- Выбрать "Сохранить как" или "Экспорт".
- Выбрать тип файла CSV (разделитель запятая) в списке доступных форматов.
- Указать имя файла и выбрать место сохранения, нажать "Сохранить".

И в итоге получается датасет обработанных значений, которые будут использованы для обучения искусственной нейронной сети. Поскольку, используемая версия Excel русскоязычная после сохранения требуется постредакция полученного датасета, например в WordPad. При помощи комбинации «ctrl+N» запятые как разделитель дробной и целой части заменяется

на точку, а точка с запятой изменяется на запятую, а также объявляются заголовки переменных в первой строке, если они не были сделаны ранее.

3.2.3. Выбор типа нейронной сети для реализации системы электронного дифференциала

Основная цель машинного обучения (ML) заключается в изучении точных систем принятия решений и разработке предикторов, способных автоматизировать задачи, которые ранее требовали человеческого вмешательства. С появлением машинного обучения были разработаны и внедрены различные алгоритмы, которые уже продемонстрировали значительные успехи в самых разных областях науки и промышленности. В частности, методы на основе ядер (kernel methods) продолжают оставаться одними из ключевых инструментов ML, показывая высокую эффективность при решении задач классификации и регрессии. [13, 46, 55, 72, 73, 110].

Поскольку ML все чаще используется в реальных приложениях, сложилось общее мнение, что одной только высокой точности прогнозирования на практике может быть недостаточно [21, 70, 108]. Вместо этого при практическом проектировании систем важнейшими характеристиками, которые обычно считаются выходящим за рамки превосходного прогнозирования, являются:

- 1) устойчивость системы к артефактам измерения или возмущениям [116];
- 2) устойчивость к дрейфу распределения данных [42];
- 3) аспекты безопасности и защищенности [12, 20, 63, 128];
- 4) требования законодательства или соблюдение социальных норм [47, 50];
- 5) способность дополнять человеческий опыт при принятии решений [57];
- 6) возможность раскрыть пользователю интересные корреляции, которые он обнаружил в данных [65, 112];
- 7) способность точно оценивать достоверность собственных предсказаний [107].

Проведённый анализ различных архитектур нейронных сетей для системы электронного дифференциала (СЭД) показал, что каждая из них обладает своими особенностями и ограничениями.

Перцептрон, являющийся сетью прямого распространения и одной из самых ранних форм нейронных сетей, был разработанный Розенблаттом [104], несмотря на свою простоту, не подходит для реализации системы СЭД из-за недостаточной гибкости.

Глубокие нейронные сети (DNN), напротив, представляют собой искусственные нейронные сети с несколькими скрытыми слоями между входным и выходным слоями (рисунок 3.11). DNN способны моделировать сложные нелинейные зависимости, что делает их универсальными для решения широкого круга задач [119, 103]. Это архитектуры, которые подходят для целей предсказания и обучения на числовых данных, как показано в главе 4 данной работы. Следовательно, глубокие нейронные сети могут быть использованы в качестве базовой архитектуры для системы СЭД.

Рекуррентные нейронные сети (RNN) предназначены для обработки данных с временными зависимостями. Они способны учитывать контекст и последовательность данных, что делает их полезными в задачах, связанных с временными рядами [71, 113, 139]. Такие сети могут применяться в будущих версиях системы СЭД, где будут учитываться дополнительные факторы, такие как погодные условия или поведение водителя, но это требует экспериментального подтверждения.

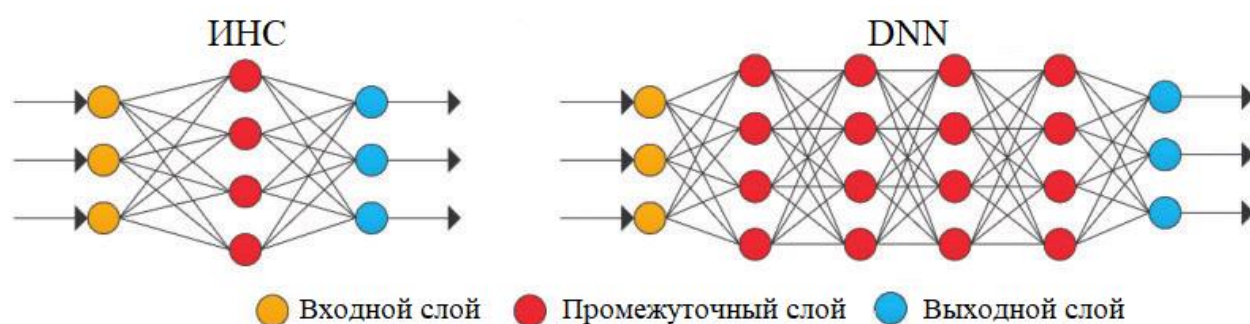


Рисунок 3.11 – Схема нейросети глубокого обучения в сравнении с обычной нейронной сетью

Сверточные нейронные сети (CNN), широко применяемые в задачах классификации изображений и видео, а также в рекомендательных системах и обработке естественного языка [10, 23, 122], не подходят для системы СЭД ввиду своей специализации на работе с визуальными данными. По той же причине

большие языковые модели [61] и случайные векторные функциональные сети связей (RVFL) [96] не подходят для данного применения.

Однослойные архитектуры, такие как RVFL и радиально-базисные функциональные нейронные сети (RBFNN) [134], хотя и эффективны в решении задач интерполяции в многомерных пространствах, обладают ограниченными возможностями и не подходят для решения сложных задач управления в системе СЭД, требующих высокой гибкости и адаптивности.

Таким образом, глубокие нейронные сети (DNN) представляются наиболее подходящей архитектурой для создания системы электронного дифференциала. В будущем возможно использование рекуррентных нейронных сетей (RNN) для более сложных версий СЭД, что потребует дополнительных исследований и тестирования.

3.2.4. Разработка программного обеспечения для обучения нейронной сети на основе полученных экспериментальных данных

В рамках данной работы будет использоваться фреймворк «TensorFlow» и его API «Keras», архитектура разработанной сети представлена на рисунке 3.12. На рисунке 3.12 входной слой (input layer) состоит из ряда независимых переменных: X_0 – угол поворота сервопривода (тяг руля, град), X_1 – скорость движения модели электромобиля, X_2, X_3, X_4 – угловое ускорение по оси «x», «y» и «z» соответственно, X_5, X_6, X_7 – ускорение свободного падения по оси «x», «y» и «z» соответственно, X_8 – температура окружающей среды, X_9 – отклонение от идеальной траектории. Далее идут несколько промежуточных полносвязанных слоёв нейронной сети глубокого обучения с указанным количеством узлов, слой dropout (предназначен для уменьшения переобучения сети), а после, на выходе (output layer) находятся 2 зависимые переменные: Y_0 – скорость вращения первого электродвигателя и Y_1 – скорость вращения второго электродвигателя.

На основе представленной архитектуры нейронной сети для системы электронного дифференциала (СЭД) электромобиля и программной реализации, можно сделать её математическое описание. Введем обозначения и опишем вычисления для каждого слоя.

Предположим, что входные данные $\mathbf{X}=[x_0, x_1, \dots, x_9]$ являются вектором размерности 10×1 .

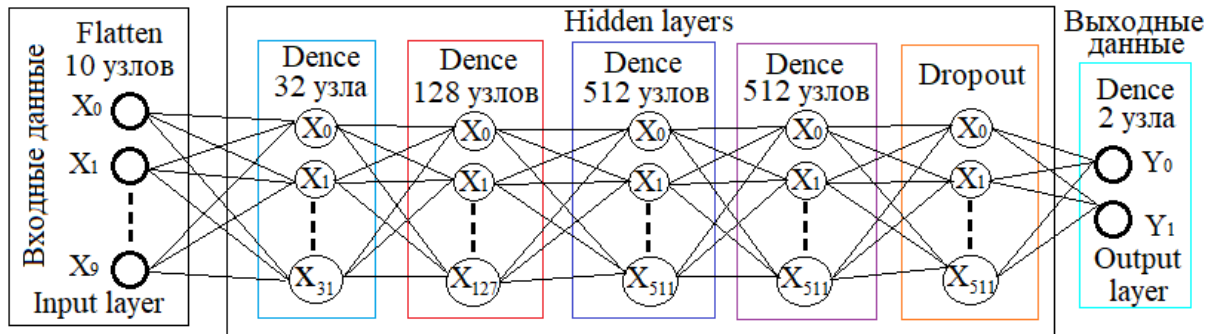


Рисунок 3.12 – Архитектура разработанной нейронной сети глубокого обучения

В первый слой входит операция *Flatten*, которая преобразует многомерный входной вектор в одномерный. В нашем случае входные данные уже одномерны, поэтому их размерность остается неизменной, поэтому $\mathbf{X} \in \mathbb{R}^{10}$. Далее следуют четыре полносвязных слоя, каждый из которых можно описать следующим образом:

1. *Первый Dense слой:*

$$\mathbf{Z}_1 = \sigma(\mathbf{W}_1 \mathbf{X} + \mathbf{b}_1), \mathbf{W}_1 \in \mathbb{R}^{32 \times 10}, \mathbf{b}_1 \in \mathbb{R}^{32}, \quad (3.1)$$

где $\mathbf{Z}_1 \in \mathbb{R}^{32}$ – это результат первой трансформации. Применяется нелинейная функция активации ReLU, которая задается как $\sigma(z) = \max(0, z)$;

\mathbf{W}_1 – матрица весов первого слоя;

\mathbf{b}_1 – вектор смещений. Размерность вектора смещений соответствует числу нейронов в слое (32 узла).

2. *Второй Dense слой:*

$$\mathbf{Z}_2 = \sigma(\mathbf{W}_2 \mathbf{Z}_1 + \mathbf{b}_2), \mathbf{W}_2 \in \mathbb{R}^{128 \times 32}, \mathbf{b}_2 \in \mathbb{R}^{128}, \quad (3.2)$$

где $\mathbf{Z}_2 \in \mathbb{R}^{128}$ – это результат второго слоя, функция активации ReLU;

\mathbf{W}_2 – матрица весов второго слоя;

\mathbf{b}_2 – вектор смещений 2 слоя, размерность равна 128.

3. *Третий Dense слой:*

$$\mathbf{Z}_3 = \sigma(\mathbf{W}_3 \mathbf{Z}_2 + \mathbf{b}_3), \mathbf{W}_3 \in \mathbb{R}^{512 \times 128}, \mathbf{b}_3 \in \mathbb{R}^{512}, \quad (3.3)$$

где $\mathbf{Z}_3 \in \mathbb{R}^{512}$ – это результат третьего слоя, функция активации ReLU;

\mathbf{W}_3 – матрица весов третьего слоя;

b_3 – вектор смещений 3 слоя, размерность равна 512.

4. *Четвертый Dense слой:*

$$Z_4 = \sigma(W_4 Z_3 + b_4), W_4 \in R^{512 \times 512}, b_4 \in R^{512}, \quad (3.4)$$

где $Z_4 \in R^{512}$ – это результат четвертого слоя, функция активации ReLU;

W_4 – матрица весов четвертого слоя;

b_4 – вектор смещений 4 слоя, размерность равна 512.

5. *Dropout слой*

Затем применяется операция Dropout, которая с вероятностью 35% случайным образом "обнуляет" часть нейронов для предотвращения переобучения. Этот слой не имеет обучаемых параметров и просто обнуляет часть значений Z_4 .

6. *Выходной слой (Dense)*

$$Y = W_5 Z_4 + b_5, \quad W_5 \in R^{2 \times 512}, b_5 \in R^2, \quad (3.5)$$

где $Y \in R^{512}$ – выходное значение, представляющее два узла выходного слоя;

W_5 – матрица весов пятого слоя;

b_5 – вектор смещений 5 слоя, размерность равна 2.

Общая формула модели

Таким образом, математическая модель нейронной сети может быть записана в виде последовательного применения линейных и нелинейных преобразований:

$$Y = W_5 \sigma(W_4 \sigma(W_3 \sigma(W_2 \sigma(W_1 X + b_1) + b_2) + b_3) + b_4) + b_5 \quad (3.6)$$

где $\sigma(z) = \max(0, z)$ – функция активации ReLU.

Векторы смещений b_i добавляют к каждому выходу слоя постоянное значение, что позволяет модели учесть случаи, когда все входные значения малы или равны нулю. Это улучшает способность сети находить оптимальные решения.

Веса W_1, W_2, W_3, W_4, W_5 – это обучаемые параметры сети, которые играют ключевую роль в преобразовании входных данных на каждом слое. Каждый набор весов представляет собой матрицу, которая отвечает за линейное преобразование данных от одного слоя к следующему. Они формируют связи между нейронами, и корректировка этих весов на основе ошибок предсказаний является центральной задачей в обучении нейронной сети. Чем больше значение конкретного веса, тем большее влияние соответствующий входной сигнал оказывает на итоговый выход

нейрона. Изначально веса инициализируются случайными значениями, после чего их корректировка происходит на основе данных и целевой функции.

Коррекция весов осуществляется через алгоритм обратного распространения ошибки. Это делается с помощью цепного правила дифференцирования:

$$\frac{\partial L}{\partial W_i} = \frac{\partial L}{\partial Y} \cdot \frac{\partial Y}{\partial W_i} \quad (3.7)$$

где L – это функция потерь (MSE, кросс-энтропия и т.д.). Градиенты показывают, насколько изменение каждого веса изменяет общую ошибку сети.

После того как градиенты вычислены, происходит обновление весов на основе выбранного оптимизатора (например, SGD, Adam и т.д.). На каждом шаге обучения веса обновляются следующим образом:

$$W_i^{new} = W_i^{old} - \eta \frac{\partial L}{\partial W_i} \quad (3.8)$$

где η – это скорость обучения (learning rate);

$\frac{\partial L}{\partial W_i}$ – градиент ошибки по весу W_i .

Основным использованным фреймворком при разработке модели НС был TensorFlow, разработанный и поддерживаемый Google. Создание нейронной сети глубокого обучения с использованием TensorFlow включает в себя несколько этапов. Общий процесс создания нейросети с использованием высокоуровневого интерфейса Keras, который теперь является частью TensorFlow:

Шаг 1: Установка TensorFlow. Данный шаг требуется при работе на локальном компьютере, при работе в Google.Colab это действие не требуется, достаточно только импортировать нужные зависимости.

Шаг 2: Импорт необходимых дополнительных библиотек.

Шаг 3: Подготовка данных. Данные были ранее собраны в csv-датасете.

Шаг 4: Определение архитектуры модели. В данной работе была выбрана нейросеть глубокого обучения архитектура определяется в том числе и гиперпараметрами (их влияние будет описано далее).

Шаг 5: Компиляция модели. Здесь определяется метрика для обучения модели.

Шаг 6: Обучение модели.

Шаг 7: Оценка и использование модели.

Полученные результаты обучения и процесс оптимизации модели нейронной сети при помощи параметров и гиперпараметров приведены в 4-й главе данной диссертационной работы.

Выводы по главе 3

1. Для концепции электромобиля с 2-мя независимыми электродвигателями сзади в данной главе были разработаны функциональная и принципиальная схемы модели электроавтомобиля, причём функциональная схема была использована в качестве опоры для принципиальной. На основе принципиальной схемы была разработана физическая модель электромобиля, которая позволит провести экспериментальное исследование и получить данные для последующего обучения НС электронного дифференциала электроавтомобиля.

2. Важным аспектом научной новизны в данной работе является – методика обучения ИНС электронного дифференциала на основе карт идеальных кривых поворота. Ключевым моментом для обучения ИНС является фиксация отклонения от этой идеальной траектории с записью данных с датчиков в этот момент.

3. В данной работе датасет для обучения нейронной сети будет иметь вид csv-таблицы со значениями, полученных с датчиков. Предварительно считанные и отфильтрованные значения данных записываются в промежуточный excel-файл для последующей обработки. После первичной обработки (фильтрация и создание структуры данных) происходит масштабирование показаний для более корректного процесса обучения ИНС.

4. Нейронные сети глубокого обучения отлично подходят для решения нелинейных систем, могут учитывать незначительные аспекты и зависимости, а также просты в обучении и могут быть использованы в качестве ядра системы СЭД.

ГЛАВА 4. ПОЛУЧЕНИЕ ЭКСПЕРИМЕНТАЛЬНЫХ ДАННЫХ ДЛЯ ОБУЧЕНИЯ НС И НАСТРОЙКА ЕЁ АРХИТЕКТУРЫ

Структурная схема разработанной системы электронного дифференциала на основе нейронной сети для управления электроавтомобилем во время поворотов и манёвров представлена на рисунке 2.8.

4.1. Получение экспериментальных данных для обучения НС с помощью физической модели электроавтомобиля

Как упоминалось ранее, методика для получения данных для нейронной сети включает наличие специальной карты идеальных кривых поворота. Подробный принцип построения и методика измерения отклонения от идеальной траектории описаны в разделе 3.2.1. На рисунке 4.1 приведено фото данной карты поворотов. Карта выполнена на листах масштабной-координатной миллиметровой бумаги для упрощения расчётов отклонения полученной траектории от идеальной. Модель электроавтомобиля (рисунок 4.2) движется по полю и рисует пройденную ей траекторию при помощи ручки со стираемыми чернилами.

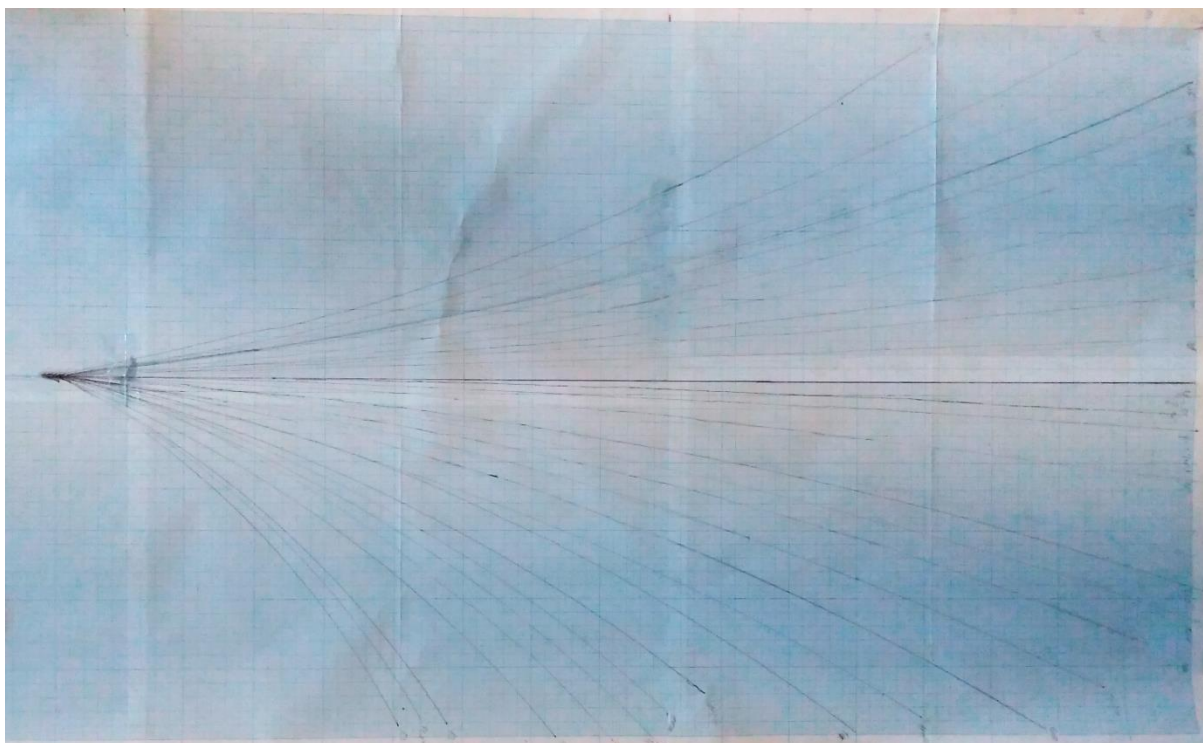


Рисунок 4.1 – Карта идеальных траекторий поворота для получения экспериментальных данных необходимых для обучения НС

Перед началом движения модели электромобиля устанавливается угол поворота сервопривода (угла поворота руля) и скорость тяговых электродвигателей. Во время движения модели (рисунок 4.3) на компьютер передаются текущие значения показаний датчиков для последующей их обработки. Подробное описание способа передачи полученного пакета данных от микроконтроллера к компьютеру приведены в разделе 3.1.

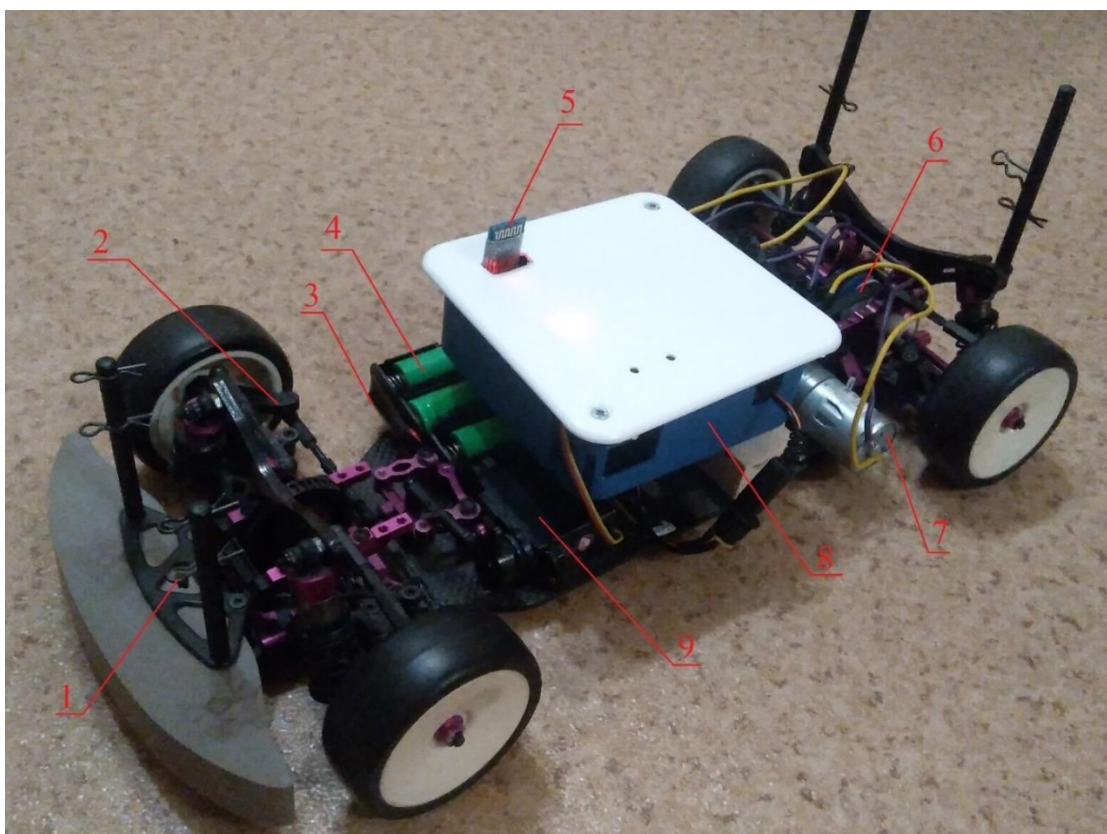


Рисунок 4.2 – Общий вид модели электромобиля, где: 1 – Отверстие для ручки; 2 – Передние тяги поворота колёс; 3 – BMS тяговой батареи; 4 – Тяговая батарея; 5 – Bluetooth-модуль для связи с ПК; 6 – Механизм передачи; 7 – Тяговые двигатели; 8 – Корпус системы управления с макетной платой внутри; 9 – Сервопривод для поворота тяг.

Получение экспериментальных данных с датчиков происходит в процессе движения модели в полуавтоматическом режиме – оператор задает момент времени для начала и окончания процесса, а также даёт команду на экспорт полученных данных от микроконтроллера к компьютеру в «excel» для дальнейшей обработки. Полученные экспериментальные данные частично отображены в таблице 4.1,

которая состоит из 217 отфильтрованных замеров, которые являются основой для датасета. Важно отметить, что для лучшей работы все данные должны быть смасштабированы в диапазон $[-1;1]$ для более корректного вычисления градиентов ошибки отклонения от истинного значения функции. Смасштабированные данные приведены в таблице 4.2, коэффициенты масштабирования подбирались при помощи вычисления наибольшего значения ($V1$, $V2$, $AngleServo$ и отклонения) и по максимально возможному кодированному 16-битному значению датчика гироскопа и ускорения (подробнее см. раздел 3.1).

В таблице 4.1 « $V1$ » – скорость вращения первого электродвигателя (заднего левого от 0 до 255); « $V2$ » – скорость вращения второго электродвигателя (заднего правого от 0 до 255), $AngleServo$ – угол поворота сервопривода (руля, от 20 до 160 град), « V » – скорость движения модели электроавтомобиля (от 0 до 255, что соответствует от 0 до 100%), « ax , ay , az » – угловое ускорение по оси « x », « y » и « z » соответственно (от -32678 до 32678, что соответствует диапазону от -250 град/с до 250 град/с), « gx , gy , gz » – ускорение свободного падения (от -32678 до 32678, что соответствует диапазону от -2g до 2g) и температуру « $temp$ ».

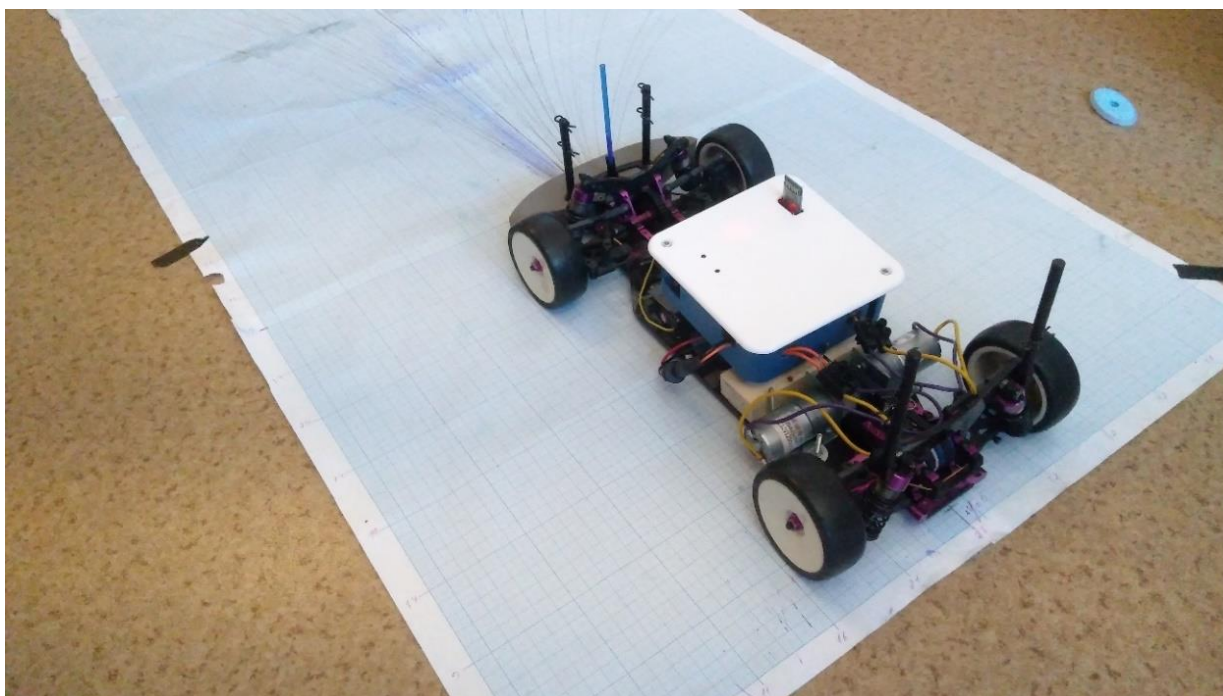


Рисунок 4.3 – Общий вид модели электроавтомобиля при проведении экспериментов по получению данных для обучения НС

Таблица 4.1 – Экспериментальные данные полученные с датчиков модели электромобиля.

V1	V2	AngleServo	V	ax	ay	az	gx	gy	gz	temp	Отклонение, град
197	197	92	197	4064	-1688	14333	-515	-515	-515	-1746	0
195	195	92	195	1305	-258	-258	-258	-258	-1	-1744	0
192	192	92	192	423	-1029	-1029	-1029	1029	-1	-1759	0
188	188	92	188	-92	-1572	19548	-582	5	-22	-1744	0
...
209	200	108	205	-900	18048	-2640	63	-1	-1	-1529	-0,3
230	219	108	225	-944	-1000	18168	-515	-515	-515	-1527	-0,4

Таблица 4.2 – Смасштабированные значения датасета в диапазоне [-1;1]

AngleServo	V	ax	ay	az	gx	gy	gz	temp	Deviation	V1	V2
0,8519	0,7725	0,124	-0,0515	0,4374	0,0157	0,0157	0,0157	0,9407	0	0,788	0,788
0,8519	0,7647	0,0398	-0,0079	0,0079	0,0079	0,0079	0,0000	0,9397	0	0,78	0,78
0,8519	0,7529	0,0129	-0,0314	0,0314	0,0314	0,0314	0,0000	0,9477	0	0,768	0,768
0,8519	0,7373	0,0028	-0,0480	0,5966	0,0178	0,0002	0,0007	0,9397	0	0,752	0,752
0,8519	0,7843	0,0338	-0,0073	0,0113	0,0000	0,0000	0,0079	0,9661	0	0,80	0,8
...
1,0000	0,8804	0,0288	-0,0305	0,5544	0,0157	0,0157	0,0157	0,8227	-0,1	0,92	0,876

4.2. Настройка архитектуры нейронной сети под полученные обработанные экспериментальные данные

Оптимизация модели нейронной сети (соотношение точность прогноза / время расчёта) является критически важным этапом в процессе её разработки и обучения. Изменение количества слоёв и узлов в архитектуре сети позволяет адаптировать модель к особенностям решаемой задачи, а также улучшить её способность к обобщению. Более глубокие нейронные сети могут захватывать сложные паттерны в данных, однако, при этом увеличивается риск переобучения

сети. Важно найти оптимальную конфигурацию, которая обеспечит баланс между сложностью модели и её рациональной производительностью на тестовых данных.

Кроме того, настройка гиперпараметров, таких как количество эпох обучения и размер `batch size`, существенно влияет на процесс обучения. Более высокое количество эпох может повысить точность модели, но также может привести к переобучению, если обучение не контролируется. Размер `batch size` определяет, сколько примеров данных обрабатывается одновременно, что влияет на скорость обучения и стабильность градиентного спуска. Проверка различных методов оптимизации и функций потерь позволяет найти наиболее эффективные подходы для конкретной задачи, что, в конечном счёте, способствует повышению качества модели и её способности к предсказанию. Таким образом, процесс оптимизации является ключевым для достижения максимальной эффективности нейронной сети в реальных приложениях.

4.2.1. Исследование влияния `batch size` на качество обучения нейронной сети

В контексте обучения нейронных сетей широко применяется алгоритм оптимизации, известный как стохастический градиентный спуск (SGD). Данный метод включает несколько ключевых этапов, начиная с генерации прогноза на основе текущих параметров модели. Затем прогноз сопоставляется с ожидаемыми результатами, и на основании этого сравнения вычисляется градиент функции ошибки. Полученный градиент используется для корректировки весовых коэффициентов модели, что способствует снижению ошибки предсказания. Этот процесс итеративно повторяется на множестве обучающих примеров до достижения желаемого уровня точности модели.

Гиперпараметр `batch size` (размер пакета) определяет количество выборок, обрабатываемых перед обновлением внутренних параметров модели. По завершении анализа пакета прогнозы сопоставляются с ожидаемыми значениями, и вычисляется ошибка. Эта ошибка применяется для оптимизации модели посредством алгоритма обратного распространения ошибки (`backpropagation`).

Таким образом, размер пакета напрямую влияет на точность предсказаний и время обучения нейронной сети.

Один из простейших подходов к решению данной задачи заключается в использовании большого размера пакета обучающей выборки. Однако при этом больший размер пакета может негативно сказаться на точности прогнозирования модели и замедлить сходимость алгоритма [34, 48, 58, 64, 74]. В то же время увеличение размера пакета способствует нахождению более плоских минимумов функции ошибки, что может улучшить обобщающую способность модели [33, 115].

Когда для создания одного пакета используются все обучающие образцы (samples), данный метод именуется пакетным градиентным спуском (batch gradient descent). Если пакет состоит всего из одного образца, метод называется стохастическим градиентным спуском (stochastic gradient descent). В случаях, когда размер пакета больше одного образца, но меньше общего числа обучающих данных, применяется мини-пакетный градиентный спуск (mini-batch gradient descent). Мини-пакетный градиентный спуск является гибридным подходом, который сочетает преимущества как стохастического, так и пакетного градиентного спуска. Популярные размеры мини-пакетов варьируются от 16 до 128 образцов [2, 5, 124, 125].

В большинстве работ [31, 83, 124, 125], чаще всего используются мини-пакетный градиентный спуск со значениями кратными 2 – 16, 32, 64, следует также отметить, что код библиотек, таких как «TensorFlow», оптимизирован именно под эти размеры и поэтому работает при этом с достаточно высокой скоростью.

На основе ПО, приведённого в *приложении Г*, был проведён ряд экспериментов, по исследованию влияния размера обучающей выборки на конечную точность прогнозирования, величину потерь и время обучения нейронной сети электронного дифференциала. Модель НС при этом и иные гиперпараметры оставались неизменными. Результаты эксперимента приведены в таблице 4.3, «val_loss» отображает итоговые потери на всей выборке, а «val_accuracy» итоговую точность прогнозирования построенной нейронной сети.

В идеальном случае «val_loss» (абсолютное значение между истинным значением функции и прогнозом) должно стремиться к нулю, а «val_accuracy» к единице. За 100% принята самая быстрая скорость обучения пакетного градиентного спуска. В скобках, рядом с методом указано количество, используемых образцов.

Графики точности предсказания и потерь для пакетного градиентного спуска и стохастического градиентного спуска показаны на рисунках 4.4 – 4.7.

Таблица 4.3 – Результаты исследования влияния batch size на НС

Метод	Потери (val_loss)	Точность (val_accuracy)	Время обучения, сек	%
SGD (1)	0,0021	0,5909	134,81	1411,62
BGD (195)	0,0030	0,7273	9,55	100
MBGD (2)	0,0009	0,6818	79,12	828,48
MBGD (4)	0,0016	0,7273	56,5	591,62
MBGD (8)	0,0011	0,5000	26,03	272,56
MBGD (16)	0,0042	0,7727	23,42	245,23
MBGD (32)	0,0019	0,6818	17,44	182,62
MBGD (64)	0,0056	0,6818	13,99	146,49
MBGD (128)	0,0026	0,6364	11,4	119,37

Пакетный градиентный спуск показал хорошие результаты в скорости обучения и точности прогнозирования, однако, при работе с наборами данных, содержащими несколько тысяч образцов, BGD-метод становится менее оптимальным из-за высоких требований к вычислительным ресурсам и объёму видеопамати. Использование меньших пакетов в таких ситуациях может быть более эффективным. Меньшие размеры пакетов снижают требования к памяти и вычислительным ресурсам, а также могут вводить полезный шум, что способствует улучшению обобщающей способности модели.

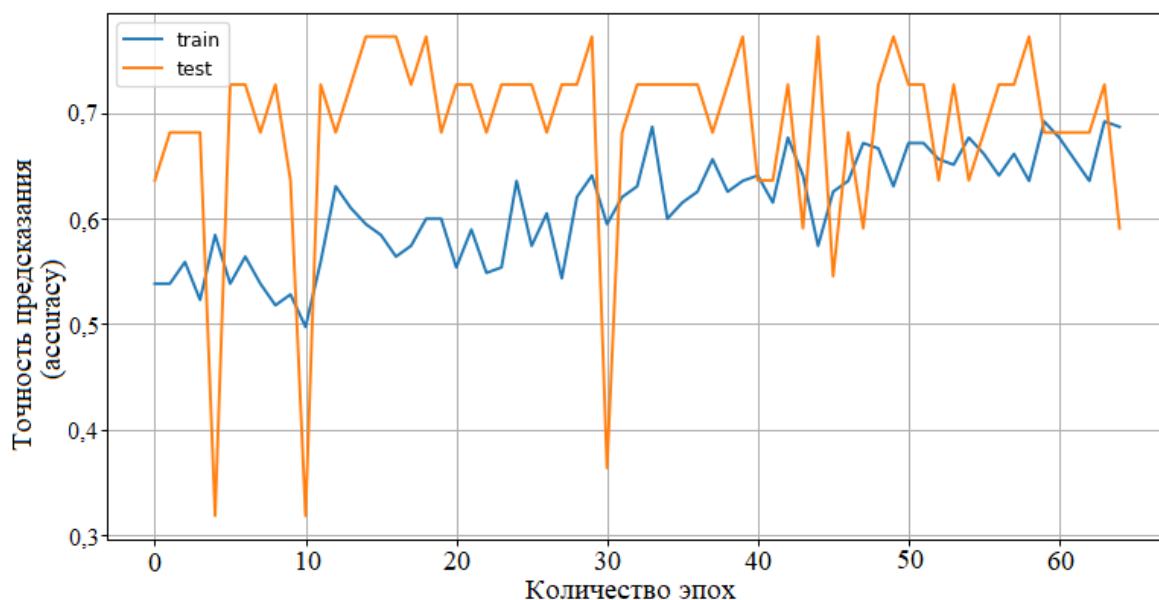


Рисунок 4.4 – График точности предсказания BGD

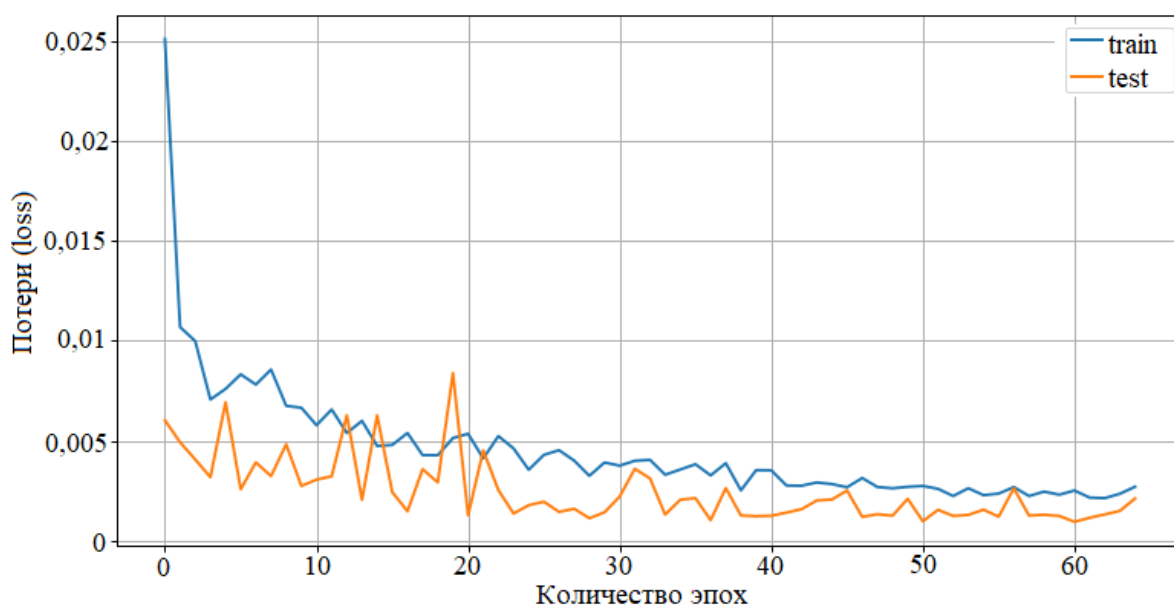


Рисунок 4.5 – График потерь BGD

Из результатов таблицы 4.3 можно увидеть, что наибольшей точностью прогнозирования обладает мини-пакетный градиентный спуск с «batch size» равным 16. Соответствующие графики точности прогнозирования и потерь приведены на рисунках 4.8, 4.9., т.к. НС при данном размере пакета показана наилучшую точность прогнозирования, то это значение было установлено в качестве конечного для разработанной модели.

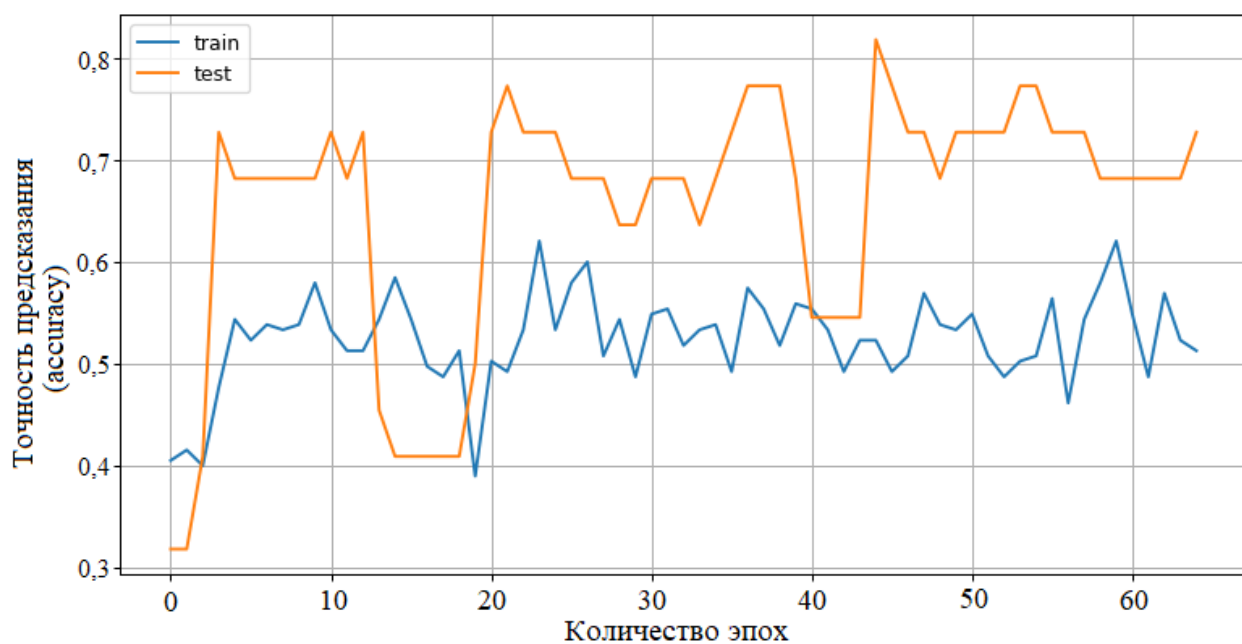


Рисунок 4.6 – График точности предсказания SGD

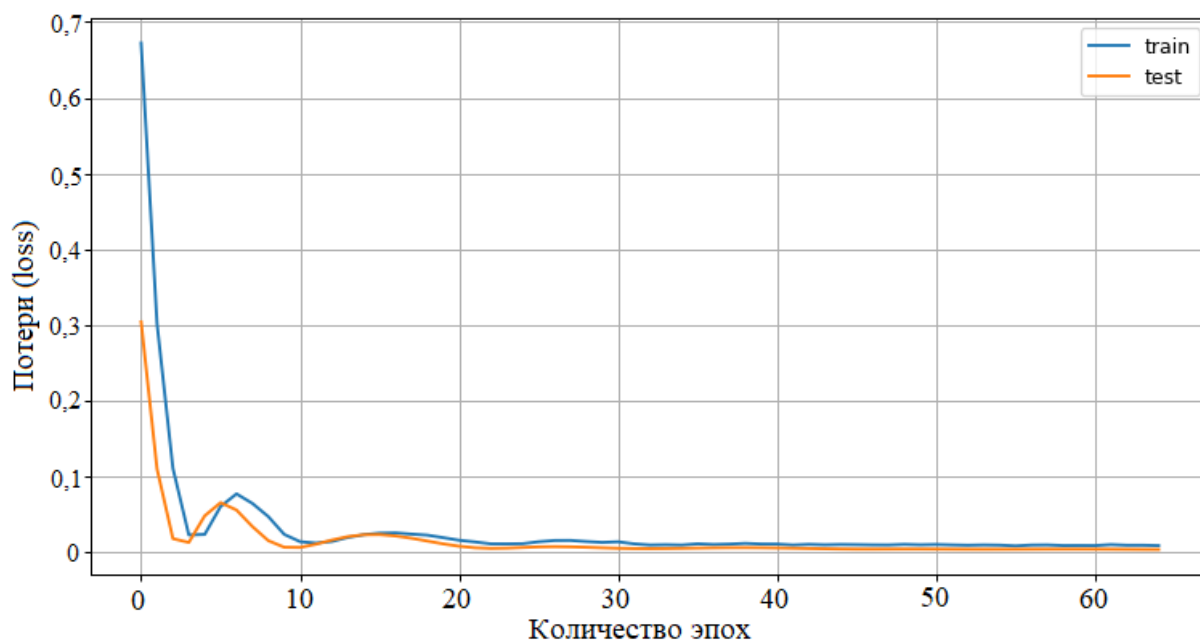


Рисунок 4.7 – График потерь SGD

Результаты эксперимента подтверждают, что размер пакета играет одну из ключевых ролей в процессе обучения искусственных нейронных сетей при использовании методов оптимизации градиентного спуска в сочетании с алгоритмом обратного распространения ошибки. Ожидаемо, наиболее эффективным методом обучения с точки зрения точности и времени оказался пакетный градиентный спуск. Однако этот метод оказывается предпочтительным лишь в случае небольших моделей и наборов данных. Более оптимальным для

большинства прикладных задач и крупных моделей оказывается мини-пакетный градиентный спуск, использующий наиболее популярные размеры пакетов – 16, 32, 64 и 128 сэмплов. Эти значения обычно являются степенями двойки, что обусловлено эффективной работой оптимизированных библиотек для матричных операций. Однако некоторые авторы работ [77, 82, 130], рекомендуют использовать размер пакета, кратный 10, для достижения высокой точности на различных наборах данных. Результаты исследования указывают на то, что лучшим выбором для разработанной модели НС является размер пакета равный 16.

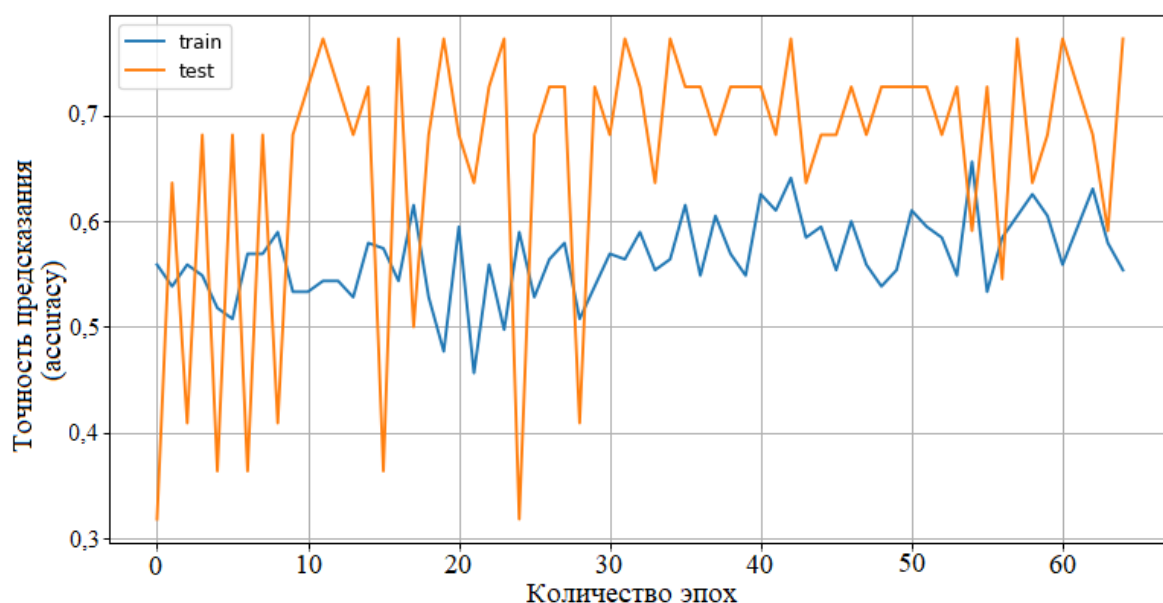


Рисунок 4.8 – График точности предсказания MBGD (размер пакета = 16)

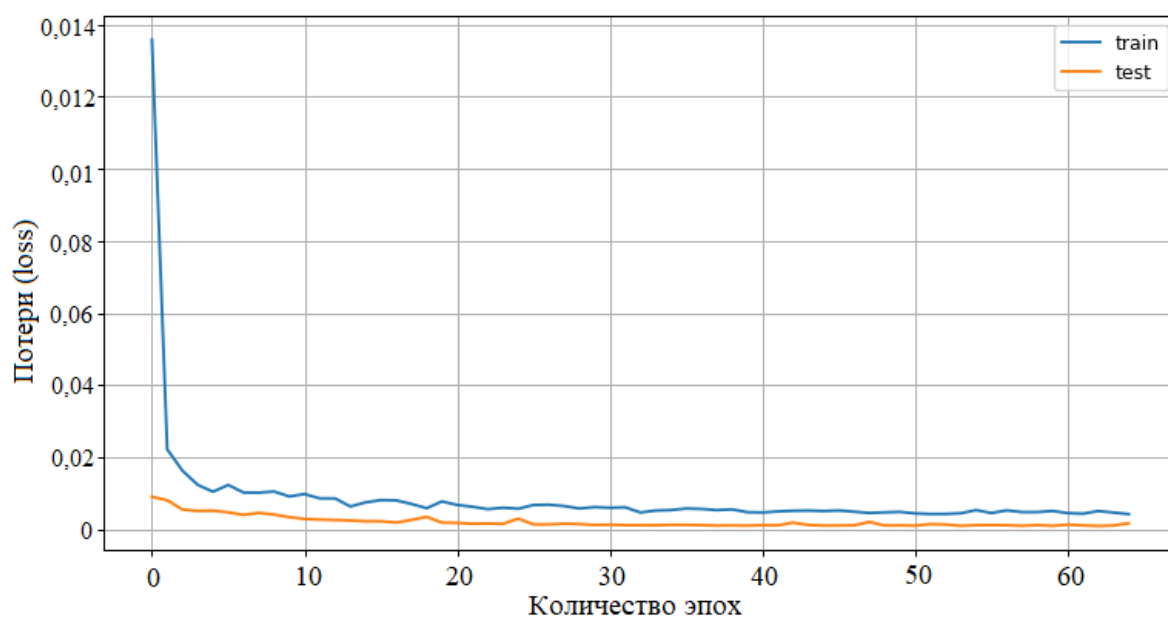


Рисунок 4.9 – График потерь MBGD (размер пакета = 16)

4.2.2. Исследование методов градиентного спуска для выбора метода оптимизации нейросети

Градиентный спуск – это способ минимизировать целевую функцию $f(x)$, параметризованную параметрами модели $x \in R$, путём обновления параметров в направлении, противоположном градиенту целевой функции $\nabla_x f(x)$ до параметров, достигая локального минимума определяется скоростью обучения [83, 106].

Среди наиболее широко используемых алгоритмов оптимизации, реализованных в библиотеках глубокого обучения, таких как Caffe, Lasagne и Keras, можно выделить следующие: стохастический градиентный спуск [109], RMSProp [35], AdaGrad [37], AdaDelta [138], Adam [68], Adamax [101] и Nadam [36, 118]. Исследователи продолжают работать над созданием новых оптимизаторов с целью достижения лучшей обобщающей способности моделей.

На основе ПО из приложения Г для обучения нейронной сети для системы электронного дифференциала был проведён ряд опытов по сравнению влияния алгоритмов градиентного спуска на конечные потери, точность прогнозирования и время обучения ИНС. Результаты приведены в таблице 4.4. Гиперпараметры и параметры НС настроены в соответствии с наилучшими вариантами из раздела ранее.

Стоит отметить, что из-за случайного выбора стартовой точки все алгоритмы, кроме «AdaMax» обладают нестабильностью конечного результата, по этой причине были проведены серии экспериментов для методов RMSProp, AdaDelta, Nadam и AdaGrad с записью наилучших результатов.

Таблица 4.4 – Сравнительный анализ алгоритмов градиентного спуска

№	Алгоритм	Время расчёта, с	val_accuracy	val_loss
1	AdaMax	9,92	0,7273	0,0013
2	RMSProp	8,69	0,5000	0,0013
3	AdaDelta	11,16	0,6818	0,3464
4	Nadam	10,36	0,7273	0,0019
5	AdaGrad	8,62	0,6818	0,0058

В результате сравнения методов градиентного спуска была выявлена одна главная особенность все алгоритмы, кроме AdaMax крайне нестабильны и могут показать как хорошие, так и очень плохие результаты. В удачном случае, алгоритмы позволят найти решение крайне быстро и эффективно, в неудачном решение потребует на порядок больше эпох обучения.

AdaGrad, несмотря на свои преимущества, имеет серьезные недостатки, особенно при обучении нейронных сетей. Хотя он эффективен в решении простых квадратичных задач, в контексте обучения нейронных сетей часто возникает проблема преждевременной остановки. Это происходит из-за того, что скорость обучения снижается настолько, что алгоритм останавливается, не достигнув глобального оптимума. Кроме того, при обучении глубоких нейронных сетей накопление квадратов градиентов с начала обучения может привести к чрезмерному снижению эффективной скорости обучения. Таким образом, AdaGrad хорошо подходит для некоторых моделей, но не является универсальным решением для всех задач глубокого обучения [69].

Преимущество AdaDelta заключается в том, что она не требует ручной настройки скорости обучения и устойчива к зашумленной градиентной информации, различным архитектурам моделей, различным модальностям данных и выбору гиперпараметров [138].

Предполагается, что RMSProp будет работать лучше в невыпуклой ситуации за счет изменения накопления градиента в экспоненциально взвешенное скользящее среднее, т.к. алгоритм использует это значение, чтобы отбросить историю из далекого прошлого. Однако, как выяснилось в результате эксперимента подход не всегда способствует более эффективной оптимизации параметров нейронной сети во время обучения.

Несмотря на то, что Nadam модифицирует компонент импульса Адама с помощью ускоренного градиента Нестерова и стремится улучшить скорость сходимости и качество изучаемых моделей [36], он также не способен выдавать стабильно хороший результат.

В качестве алгоритма оптимизации предпочтителен AdaMax, т.к. нет необходимости корректировать смещение инициализации. Этот алгоритм не является самым быстрым алгоритмом из всех ранее представленных, но при этом наиболее стабилен, что играет ключевую роль при поиске минимума сложных функций. Кроме того, величина обновления параметров имеет более простую оценку с помощью AdaMax по сравнению со стандартным Adam [68].

4.2.3. Исследование влияния функции потерь на точность прогнозирования НС

Функция потерь (loss function) в нейронных сетях представляет собой метрику, которая измеряет разницу между предсказанными значениями модели и фактическими значениями (целевыми значениями). Задача обучения нейронной сети состоит в минимизации этой функции потерь. В данном эксперименте были рассмотрены следующие функции:

- Binary Cross-Entropy [105];
- Binary Focal Cross-Entropy [81];
- Categorical Cross-Entropy;
- Categorical Focal Crossentropy;
- Mean Absolute Error [97];
- Mean Squared Error [67];
- Mean Squared Logarithmic Error;
- Squared Hinge [75].

В результате сравнения алгоритмов для минимизации потерь на основе программного обеспечения, приведенного в *Приложении Г*, были получены данные, отображённые в таблице 4.5.

Таблица 4.5 – Сравнительный анализ функций потерь

№	Алгоритм	val_loss	val_accuracy	Время обучения, с
1	Binary Cross-Entropy	0,4770	0,5909	10,31
2	Binary Focal Cross-Entropy	0,0304	0,3182	10,28
3	Categorical Cross-Entropy	0,0345	0,3182	9,71

4	Categorical Focal Crossentropy	0,0697	0,8182	19,1
5	Mean Absolute Error	0,0383	0,6818	9,70
6	Mean Squared Error	0,0020	0,8636	16,6
7	Mean Squared Logarithmic Error	0,0005	0,6818	9,74
8	Squared Hinge	0	0,2727	10,34

Как можно заключить из данных, представленных в таблице 4.5 наиболее оптимальными вариантами в задачах прогнозирования и классификации являются «Categorical Focal Crossentropy» и «Mean Squared Logarithmic Error», «Mean Squared Logarithmic Error», а худшим вариантом оказался Squared Hinge. Исходя из результатов эксперимента, для разработанной модели НС лучшим вариантами оказались: Categorical Focal Crossentropy и Mean Squared Error, соответствующие графики процесса обучения для этих алгоритмов приведены на рисунках 4.10 и 4.11.

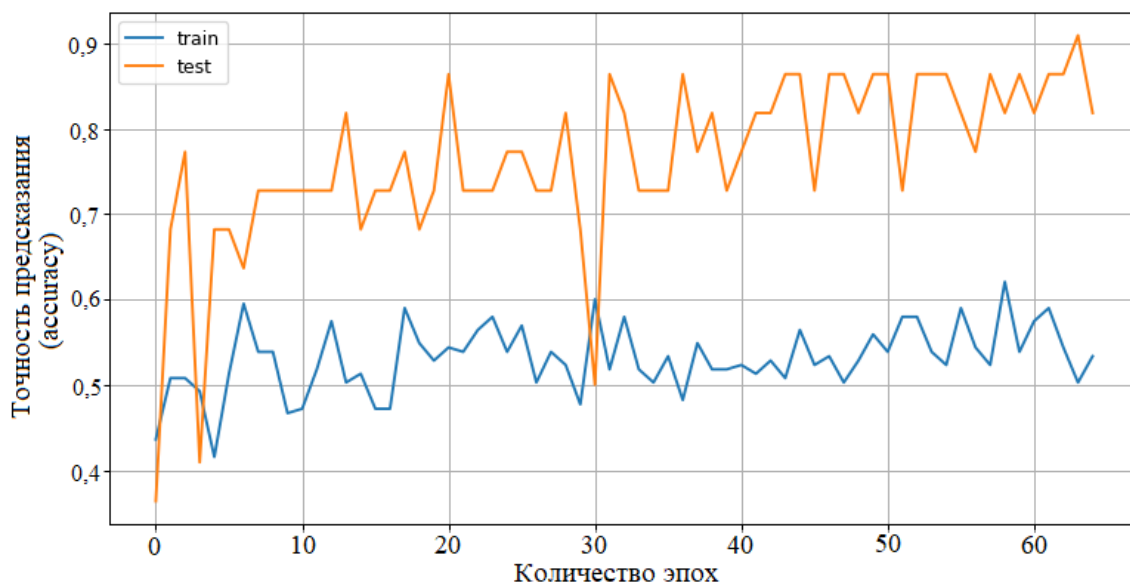


Рисунок 4.10 – График точности предсказания (Categorical Focal Crossentropy)

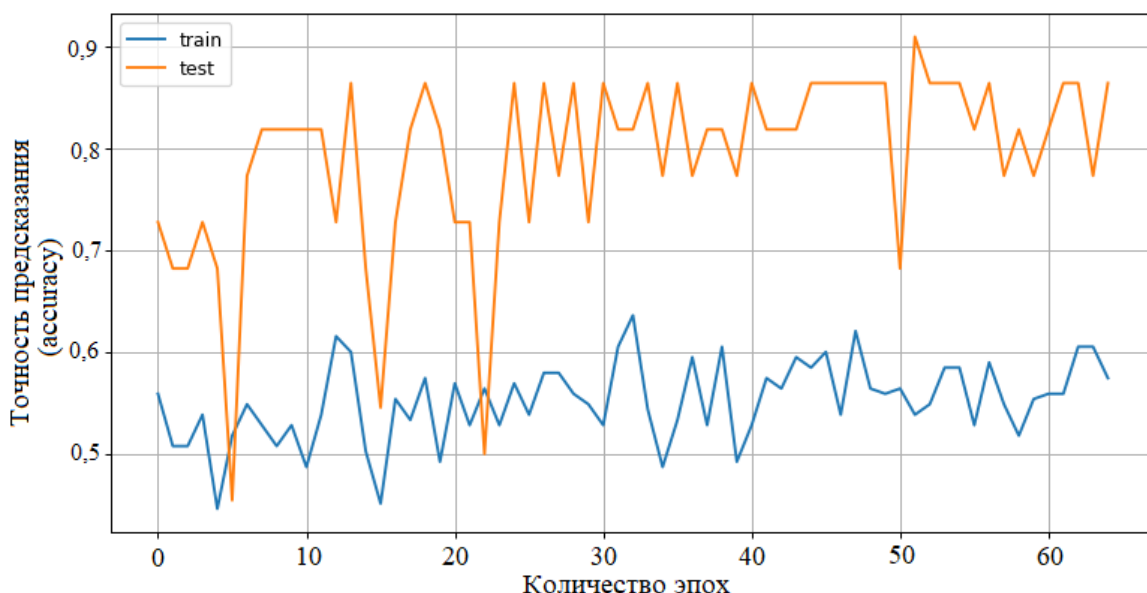


Рисунок 4.11 – График точности предсказания (Mean Squared Error)

4.3. Результаты обучения конечной архитектуры НС

Для обучения нейронной сети (*приложение Г*) данные из таблицы 4.2 были переведены в csv-формат (подробнее о преобразовании изложено в разделе 3.2.2). Для выбранного типа нейронной сети (выбор был произведён в разделе 3.2.3) – нейронная сеть глубокого обучения при помощи фреймворка «Tensorflow» и библиотеки «keras» была разработана архитектура, представленная ниже:

```
def get_basic_model():
    model = tf.keras.models.Sequential([
        tf.keras.layers.Flatten(),
        tf.keras.layers.Dense(32, activation='relu', input_shape = (10,)),
        tf.keras.layers.Dense(128, activation='relu'),
        tf.keras.layers.Dense(512, activation='relu'),
        tf.keras.layers.Dense(512, activation='relu'),
        tf.keras.layers.Dropout(0.35),
        tf.keras.layers.Dense(2)])

    model.compile(metrics = ['accuracy'], optimizer = NN_optimizer, loss = NN_loss)
    return model
```

Для данной архитектуры были выбраны следующие гиперпараметры (настройки в начале ПО, см. *приложение Г*): размер пакета (batch size) = 16; метод градиентного спуска для оптимизации нейросети – «AdaMax», количество эпох =

65 и функция потерь – «MSE» (Mean Squared Error) как наиболее оптимальная комбинация, исходя из результатов проведённых экспериментов (раздел 4.2).

Графики процесса обучения функции потерь (рисунок 4.12) и точности прогнозирования (рисунок 4.13) приведены ниже. Из данных графиков можно увидеть хорошую точность прогноза, даже при условии небольшого количества исходных данных для обучения ИС. В результате проверки работы ИС, были получены: точность прогнозирования равная 0,8636 со среднеквадратичным отклонением для тренировочной выборки 0,042, а для тестовой 0,045.

Среднее время на вычисление текущего задания скорости тягового электропривода составило $0,0213 \pm 5\%$ секунд при задании отдельными массивами (пакетами) с каждым новым вызовом функции предсказания. С единоразовым вывозом функции предсказания для тестовой выборки из 22 пакетов составило $0,0372 \pm 5\%$ секунды.

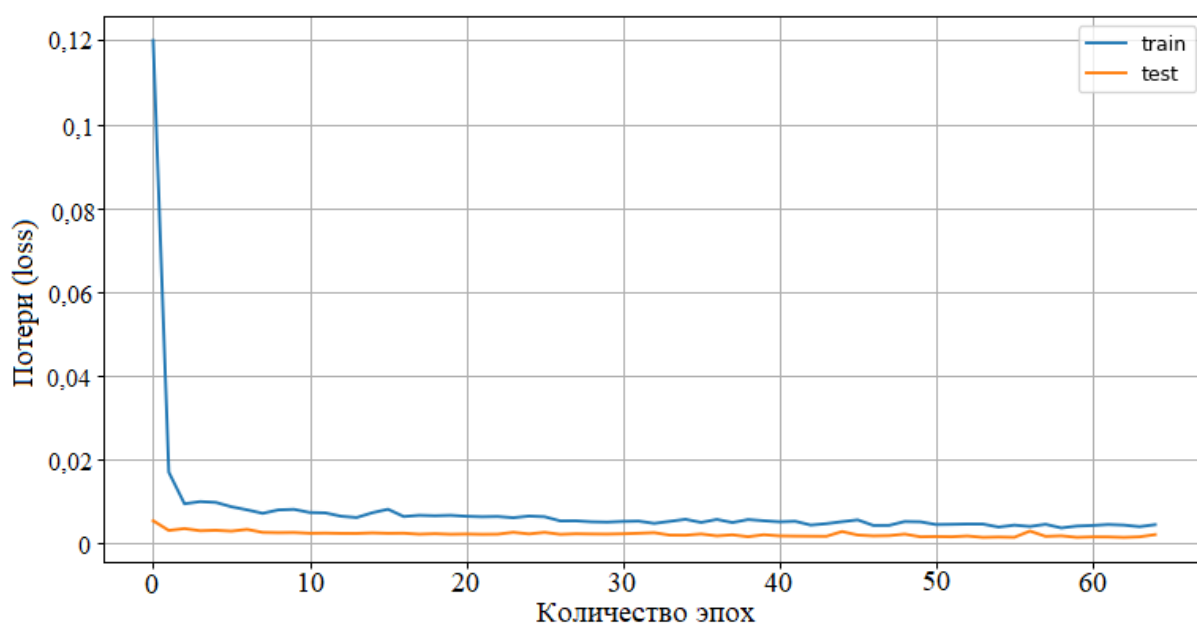


Рисунок 4.12 – График потерь (loss) в процессе обучения ИНС

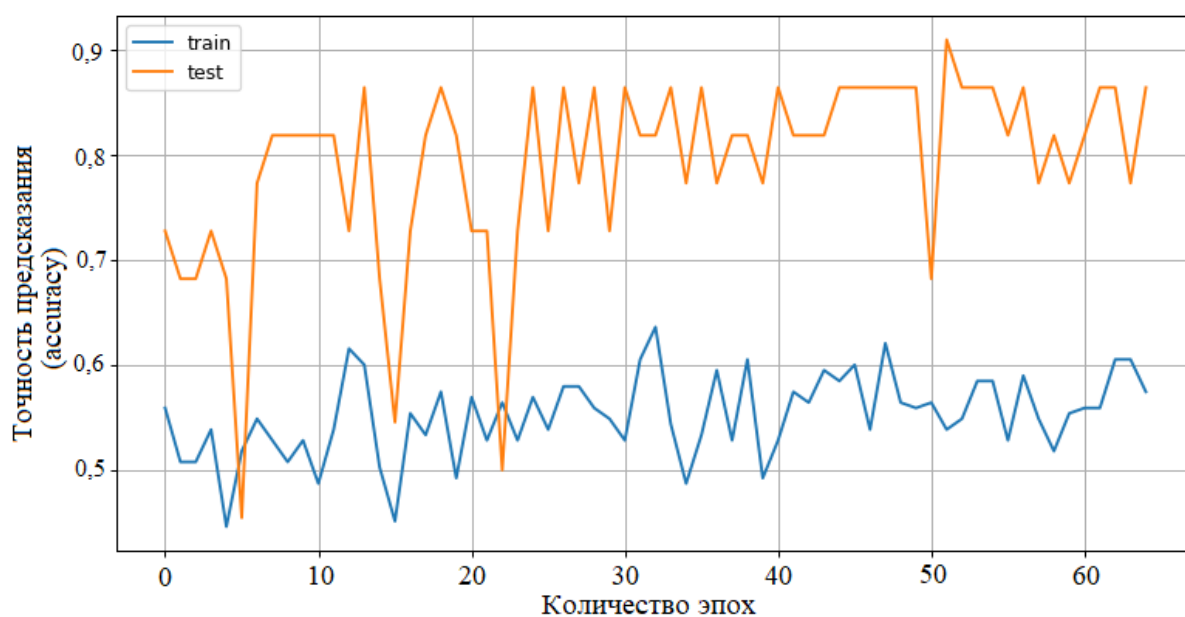


Рисунок 4.13 – График точности прогнозирования при обучении НС

Результат работы нейронной сети для системы электронного дифференциала с проверкой точности прогнозирования на тестовой выборке (не участвовала в процессе обучения модели НС) для скорости V1 и V2, приведены на рисунках 4.14, 4.15. Более наглядно точность предсказания модели нейронной сети, на нескольких примерах, тестовой выборки можно увидеть в таблице 4.6. Как упоминалось ранее, размер датасета не позволяет достичь идеального совпадения с истинным значением скорости вращения электропривода. Дальнейшее повышение качества прогнозирования возможно, но при этом необходимо учитывать значительное увеличение времени при использовании ручного подхода. Для создания датасета, используемого в наших исследованиях, понадобилось порядка 13 часов на замеры и обработку данных, поэтому более эффективным подходом является использование автоматизированной системы, о которой подробнее будет изложено далее в главе 4.4.

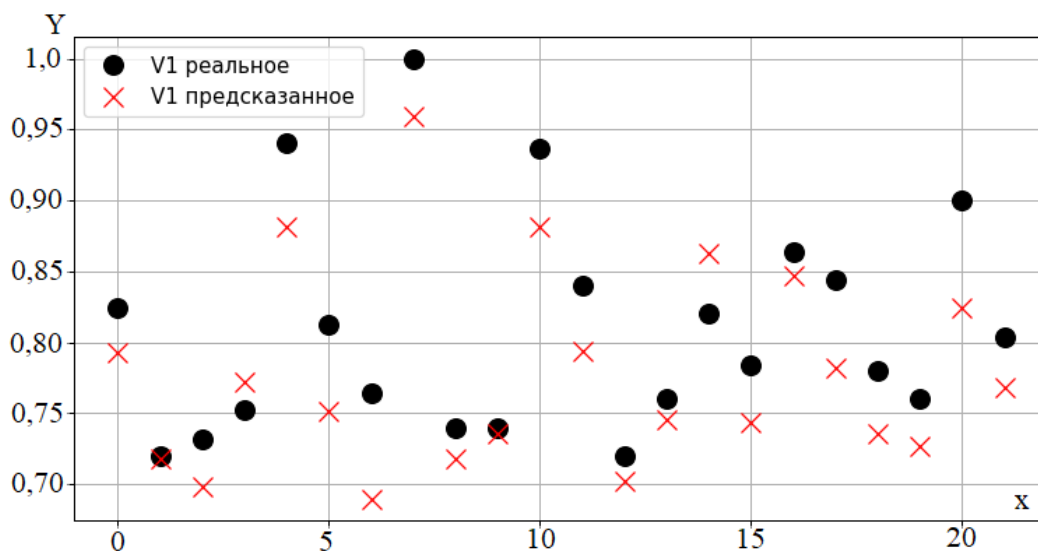


Рисунок 4.14 – Графики прогноза V1 модели нейронной сети на тестовой выборке

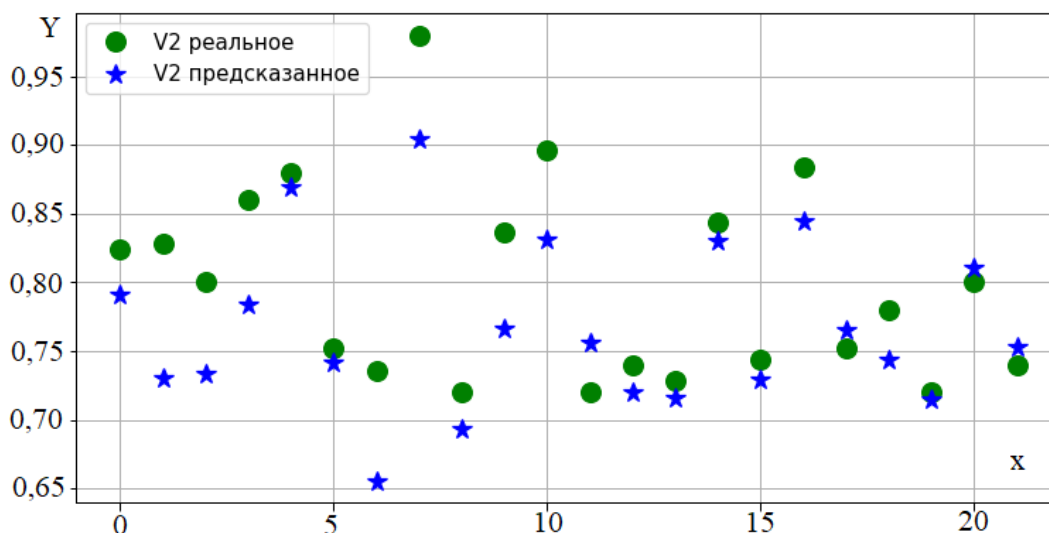


Рисунок 4.15 – Графики прогноза V2 модели нейронной сети на тестовой выборке

Таблица 4.6 – Предсказание скорости вращения электродвигателей V1, V2 модели нейронной сети на тестовой выборке

№ пакета (x)	V ₁ реальное	V ₁ предсказанное	V ₂ реальное	V ₂ предсказанное
0	0,824	0,792	0,824	0,791
1	0,72	0,718	0,828	0,730
2	0,732	0,698	0,8	0,733
3	0,752	0,772	0,86	0,784
7	1	0,959	0,98	0,904
8	0,74	0,717	0,72	0,693
9	0,74	0,735	0,836	0,766

12	0,72	0,702	0,74	0,720
14	0,82	0,863	0,844	0,830
18	0,78	0,735	0,78	0,744
19	0,76	0,727	0,72	0,715
21	0,804	0,768	0,74	0,753

Как можно видеть из рисунков 4.14, 4.15 и таблицы 4.6 НС стремится к своим образцовым значениям и иногда достигает их (1-й пакет для V1, 9-й пакет для V1, 14-й пакет для V2, 19-й пакет для V2). При этом НС часто отражает отношения скоростей между собой (разница в абсолютном выражении между V1 и V2). Полученные результаты позволяют утверждать, что разработанная система электронного дифференциала работоспособна и её можно рекомендовать к практической реализации в качестве отдельного дополнительного устройства или в составе электротехнического комплекса электроавтомобиля.

Кроме проверки системы электронного дифференциала на основе нейронной сети с использованием вычислительных методов, был проведён дополнительный сравнительный натурный эксперимент, направленный на оценку эффективности управления моделью электромобиля с существующими линейными и нелинейными системами. Результаты сравнительного анализа систем электронного дифференциала при поворотах на углы 100^0 и 120^0 при варьировании скорости от минимальной (30%), средней (50%) и максимальной представлены на рисунке 4.16. Значения отклонений для тестируемых систем при соответствующих скоростях вращения колёс представлены на рисунках 4.17 и 4.18. Результаты эксперимента показали следующее:

1. Система электронного дифференциала на основе нейронной сети демонстрирует лучшую адаптивность к изменяющимся условиям и конструктивным особенностям модели электромобиля.

2. В случае, если данные с датчиков близки или совпадают с информацией, использованной при обучении модели, электрообиль следует по идеальной траектории, в противном случае наблюдается незначительное «рысканье» в районе идеальной траектории.

3. Увеличение скорости и угла поворота ведёт к большему отклонению от идеальной траектории для линейных и нелинейных систем управления.

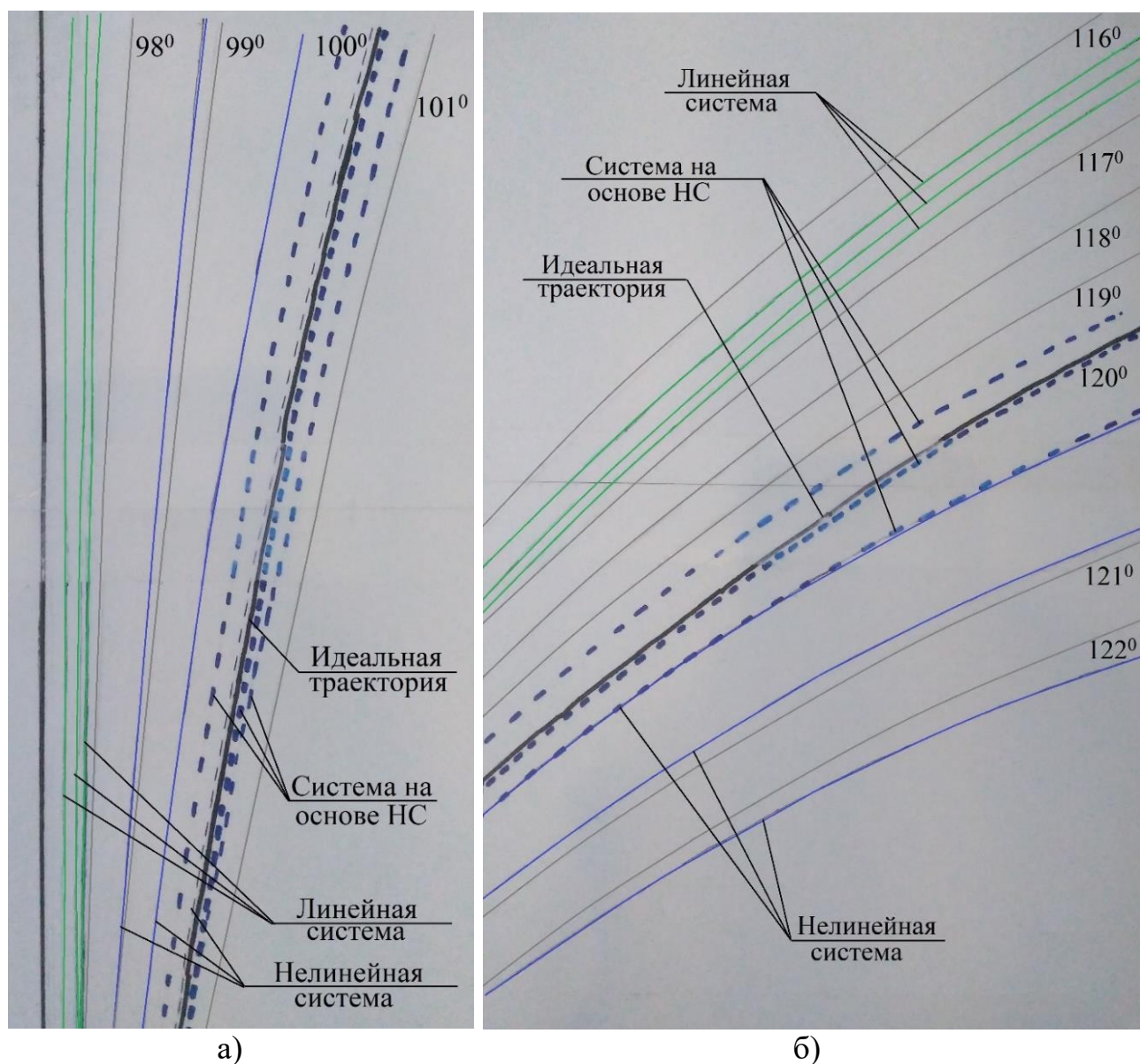


Рисунок 4.16 – Результаты сравнительного натурального эксперимента систем электронного дифференциала при повороте сервопривода а) 100° , б) 120°

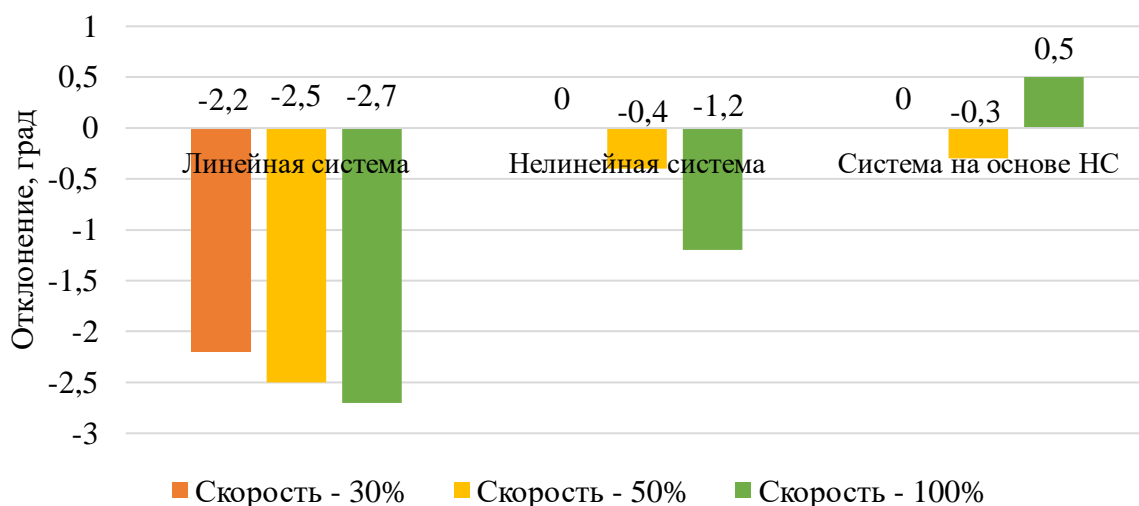


Рисунок 4.17 – Величина отклонения от идеальной траектории для линейной, нелинейной и системы на основе НС при повороте сервопривода на 100°

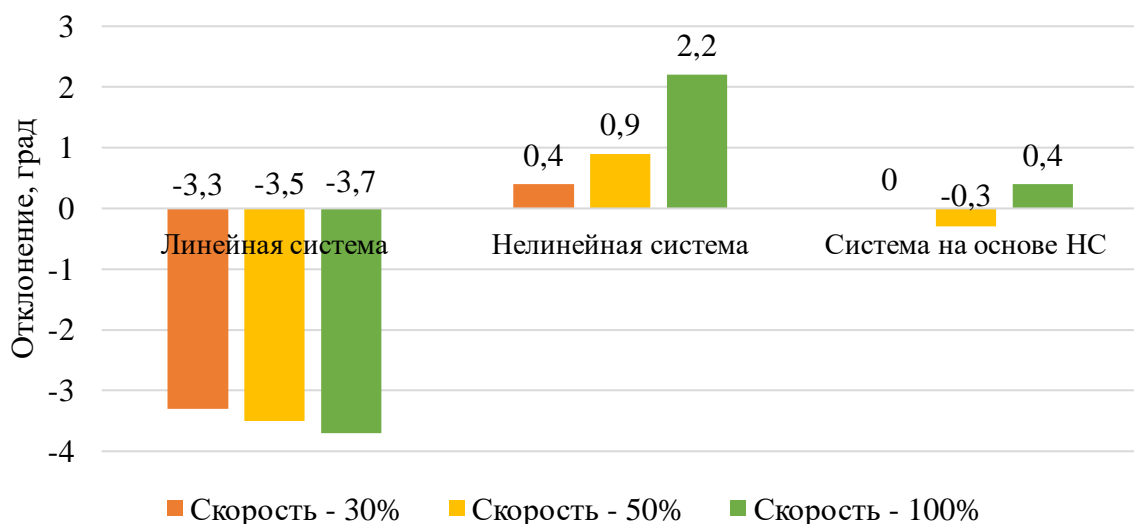


Рисунок 4.18 – Величина отклонения от идеальной траектории для линейной, нелинейной и системы на основе НС при повороте сервопривода на 120°

4.4. Перспективы дальнейших исследований

Перспектива самообучения разработанной модели СЭД на основе НС

Самообучающаяся электронная дифференциальная система (СЭД) – это важное направление в разработке интеллектуальных транспортных средств, способное повысить эффективность и безопасность электроавтомобилей. СЭД адаптируется к условиям, учитывая погодные факторы (метель, туман, ураганные условия), привычки водителя и особенности дорожного покрытия, что позволяет оптимизировать управление транспортом и предотвращать пробуксовку в сложных

ситуациях, например, при гололеде. В основе работы системы лежат алгоритмы обучения с подкреплением и поведенческого клонирования, которые позволяют накапливать и использовать данные для дальнейшей адаптации, подстраивая скорость и сцепление с дорогой в зависимости от текущих условий [11, 22 29, 40, 78, 93, 99, 114, 117, 121].

Перспективы реализации полноприводного автомобиля с четырьмя электродвигателями

Реализация полноприводного автомобиля с четырьмя электродвигателями (по одному на каждое колесо) является перспективной для беспилотных и внедорожных решений, а также автомобилей класса люкс, благодаря значительным преимуществам в управляемости и эффективности в сложных условиях эксплуатации (см. рисунок 1.11, таблицу 1.1). Для адаптации разработанной системы электронного дифференциала, основанной на нейронных сетях, требуется добавление двух дополнительных приводов на переднюю ось и доработка программного обеспечения, расширяющего контроль параметров и функций управления. Это предполагает модификацию ПО для обучения нейронной сети и использование более мощного контроллера, например STM32 или ESP32, для надежной работы многоприводной системы (см. рисунок 4.19).

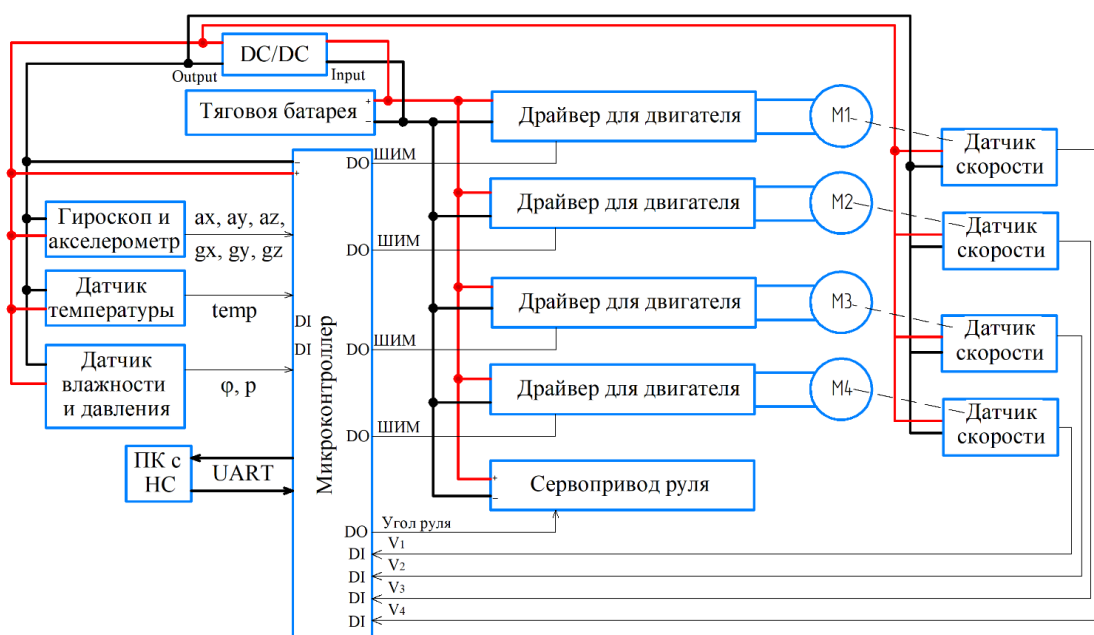


Рисунок 4.19 – Функциональная схема электронного дифференциала для 4-х двигательной модели электромобиля

Повышение адаптивности системы электронного дифференциала на основе нейронных сетей за счёт интеграции дополнительных сенсоров

Система электронного дифференциала на базе нейронных сетей показала высокую эффективность, но для повышения адаптивности и безопасности целесообразно интегрировать дополнительные датчики: освещённости, ультразвуковые и инфракрасные. Датчик освещённости позволит распознавать время суток и фиксировать плохую видимость при тумане, корректируя поведение автомобиля. Ультразвуковой датчик будет измерять дистанцию до других транспортных средств, предотвращая резкие торможения, а инфракрасный сенсор поможет избегать столкновений с пешеходами и животными. Обучение нейронной сети потребует реальных сценариев с симуляцией трафика и препятствий, используя Minmax-алгоритм, который минимизирует потери в экстремальных ситуациях, улучшая устойчивость системы [98, 100].

Упрощение процесса получения данных для обучения нейронной сети электронного дифференциала с использованием камеры и компьютера.

Для упрощения получения данных для обучения нейронной сети электронного дифференциала электромобиля можно использовать камеру, лазер и компьютер. Камера, установленная позади автомобиля, фиксирует пройденный путь (см. рисунок 4.20), а лазер отмечает траекторию движения, позволяя вычислять отклонения реальной траектории от эталонной (рисунок 4.21). Стабилизация камеры необходима для исключения искажений изображения. Такой подход позволяет не только фиксировать отдельные повороты, но и анализировать их последовательности, что способствует более устойчивому обучению. Данные можно автоматически обрабатывать с помощью методов компьютерного зрения для перевода изображения в координаты, улучшая точность и эффективность анализа [66, 143].

Антиблокировочная система и её интеграция с СЭД

Интеграция ABS с электронным дифференциалом на основе нейронных сетей позволяет создать систему, способную адаптироваться к изменяющимся условиям и повышающую безопасность управления.

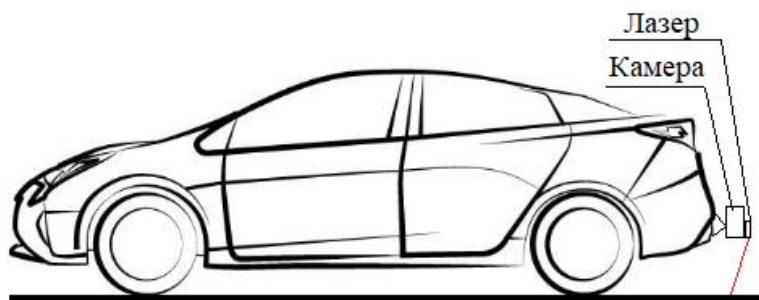


Рисунок 4.20 – Расположение камеры для съёмки траектории движения автомобиля

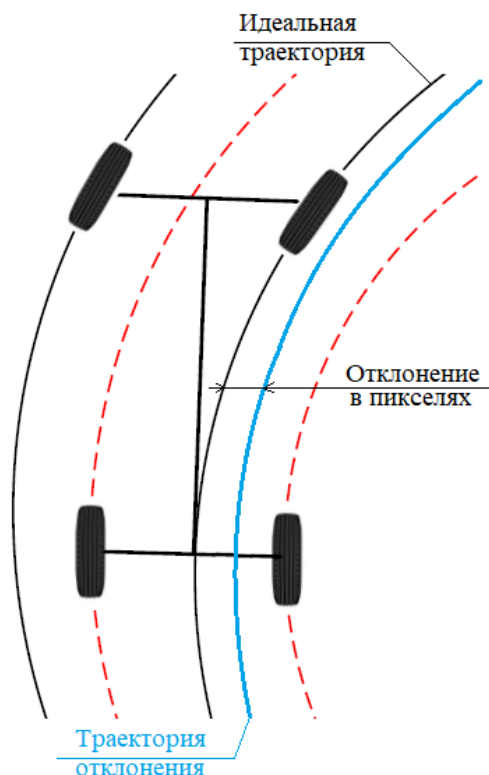


Рисунок 4.21 – Иллюстрация отклонения от идеальной траектории для его автоматического вычисления

Нейронные сети позволяют анализировать данные о торможении, скорости вращения колёс и дорожных условиях в реальном времени, обеспечивая точное перераспределение тормозных усилий и предотвращение блокировки колёс. При этом сигналы от ABS можно использовать для оптимального управления тормозными клапанами, а тяговый двигатель автоматически отключается при активации ABS для лучшего контроля торможения (см. рисунок 4.22). В этой схеме на рисунке 4.22 условно обозначены переднее левое (FL), переднее правое (FR), заднее левое (RL) и заднее правое (RR) колёса. Дополнительно система может сокращать тормозной путь, включая тяговый двигатель в режим

противовключения при экстренном торможении на скользких покрытиях, что улучшает стабилизацию и безопасность транспортного средства

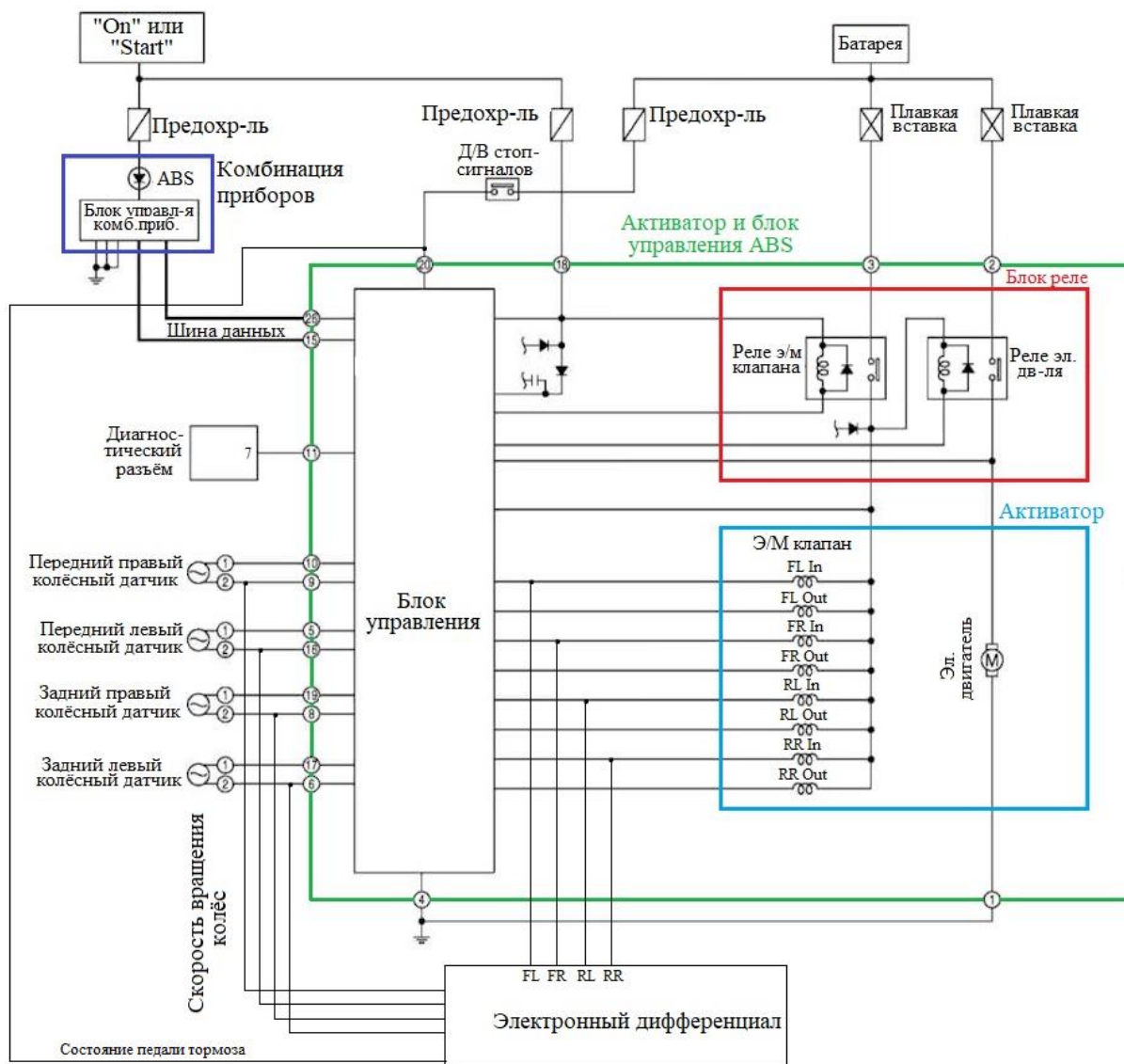


Рисунок 4.22 – Функциональная схема дополнения ABS системой электронного дифференциала

Адаптация модели НС для работы с маломощными вычислительными устройствами

Система электронного дифференциала, требующая высоких вычислительных мощностей, может быть оптимизирована следующими методами.

1. Квантизация – уменьшение разрядности чисел, переводя их с FP32 на INT8, снижает объем памяти и ускоряет вычисления, что особенно полезно для ограниченных ресурсов [44, 49, 129]. Доступны два варианта: пост-тренировочная квантизация и квантизация с дообучением.

2. Принаряжение весов – удаление малозначимых весов, что уменьшает вычислительные затраты и память. Доступны универсальное и структурное принаряжение, которое удаляет целые нейроны или фильтры [14, 26, 126].

3. Сжатие модели – включает методы, такие как кодирование весов и сингулярное разложение (SVD), что снижает объем параметров [27, 32, 79].

4. Трансформация архитектуры – использование легковесных архитектур (например, MobileNet, SqueezeNet) для уменьшения вычислений, сохраняя производительность [17, 18].

5. Оптимизированные библиотеки и фреймворки – TensorFlow Lite и другие библиотеки поддерживают методы оптимизации для мобильных устройств.

6. Факторизация матриц – разложение больших матриц весов через SVD уменьшает вычислительные операции [16, 25, 84].

7. Аппаратные оптимизации – использование нейронных процессоров на устройствах повышает эффективность исполнения модели [19].

Выводы по главе 4

1. Получены экспериментальные данные в объёме достаточном для обучения нейронной сети.

2. В ходе исследования зависимости эффективности методов обучения от размера пакета, наиболее быстрым процессом обучения продемонстрировал пакетный градиентный спуск (BGD), тогда как метод стохастического градиентного спуска (SGD) оказался наиболее медленным и малоэффективным. Максимальная точность прогнозирования достигнута при использовании мини-пакетного градиентного спуска с размером пакета, равным 16.

3. Сравнение различных методов градиентного спуска выявило, что алгоритм «AdaMax» предпочтителен в качестве метода оптимизации, поскольку его применение не требует точной настройки начальной инициализации.

4. Анализ методов минимизации функции потерь показал, что наиболее рациональной функцией является «Mean Squared Error».

5. Настойка модели НС для СЭД позволяет повысить точность прогнозирования с 49,18% до 86,36% (1,8 раза), снизить количество узлов сети с 2 млн. до 160 тыс. (в 12,5 раз), что существенно сократило требования к памяти устройства и позволило увеличить скорость расчёта с $77\pm 5\%$ мс до $23\pm 5\%$ мс (3,3 раза).

6. Результаты эксперимента показали, что нейронная сеть стремится воспроизводить целевые значения и в некоторых случаях точно повторяет их. Она также демонстрирует способность отражать взаимосвязи скоростей (разницу в абсолютных значениях между $V1$ и $V2$), что свидетельствует о работоспособности системы электронного дифференциала на основе НС.

7. Разработанную систему электронного дифференциала на основе искусственных нейронных сетей можно адаптировать для управления полноприводным транспортным средством.

8. Процесс получения данных для обучения нейронной сети электронного дифференциала электромобиля может быть автоматизирован с использованием камеры, лазера и компьютера.

9. Интеграция системы ABS с электронным дифференциалом на основе нейронных сетей создаёт перспективы для более точного и адаптивного управления автомобилем в реальном времени.

10. Оптимизация работы нейронной сети для устройств с ограниченными вычислительными ресурсами возможна за счет применения методов квантования, сжатия модели, нормализации весов, трансформации архитектуры с использованием специализированных библиотек и фреймворков, а также факторизации матриц и аппаратных оптимизаций.

ОБЩИЕ ВЫВОДЫ И РЕЗУЛЬТАТЫ

1. На основе анализа научно-технической литературы определена рациональная компоновка тягового электропривода электроавтомобиля – два независимых двигателя сзади и полноприводная система с 4-мя двигателями, т.к. данные компоновки являются наиболее рациональными с точки зрения эффективности управления и относительной дешевизны.

2. Анализ эффективности существующих систем электронного дифференциала для управления тяговым электроприводом электроавтомобиля при движении на поворотах и манёврах показал, что линейная система электронного дифференциала из-за отсутствия учёта в данной системе конструктивных особенностей транспортного средства, скорости движения и других факторов является неэффективной. Нелинейная система, в отличие от линейной, учитывает эти факторы, но при движении электроавтомобиля на скоростях более 50 км/ч эффективность управления снижается.

3. Обоснован выбор нейронной сети для системы электронного дифференциала для управления электроприводом электроавтомобиля. Использование нейронной сети в качестве ядра электронного дифференциала является рациональным решением, т.к. она предоставляет возможность учитывать дополнительные факторы, влияющие на эффективность управления электроавтомобилем при повышенных скоростях движения во время поворотов (конструктивные характеристики автомобиля, силы инерции, взаимодействие шин с дорожным покрытием, стиль вождения), а также появляется возможность интегрировать дополнительные системы стабилизации и устойчивости автомобиля (ABS, ESP и т.д).

4. Разработана физическая модель электрического автомобиля, позволяющая получать экспериментальные данные, необходимые для последующего обучения нейронной сети.

5. Разработана методика получения экспериментальных данных необходимых для обучения нейронной сети, учитывающей отклонение от идеальной кривой поворота. Предложенный метод легко масштабируется, поддается автоматизации

и не требует дополнительных действий при изменении количества измеряемых параметров.

6. Выбрана нейронная сеть глубокого обучения для системы электронного дифференциала для управления электроавтомобилем, осуществлено обучение НС на основе экспериментальных данных, полученных с помощью физической модели электроавтомобиля. В результате обучения НС СЭД были получены: точность прогнозирования задающего значения скорости вращения тяговых электродвигателей равная 86,36% и скорость расчёта прогнозов $23 \pm 5\%$ мс.

7. Сравнение линейной и нелинейной СЭД с СЭД на основе НС показало отклонение от идеальной траектории поворота на 0,4 градуса, что значительно лучше по сравнению с линейной (3,7 градуса) и нелинейной (2,2 градуса).

8. Использование системы электронного дифференциала на основе нейронных сетей для электроавтомобилей и беспилотных наземных транспортных средств позволяет повысить безопасность транспорта на дорогах.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Бутенко В. Н., Мосин М. Е., Чернов И. В. Электронный дифференциал электромобиля с полным приводом // наука. Технологии. Инновации. – 2020. – С. 143-146.
2. Возмилов А. Г., Лисов А. А., Урманов В. Г., Синева Г. Н.: Определение вида заболеваний, поражающих листья картофеля, с использованием алгоритмов машинного обучения // Вестник НГИЭИ. 2023. № 3 (142). С. 7–16. DOI: 10.24412/2227-9407-2023-3-7-16
3. Возмилов А.Г, Урманов В.Г, Лисов А.А, Илимбетов Р.Ю. Разработка и моделирование пропорционально-интегрального регулятора для электронного дифференциала электротрактора сельскохозяйственного назначения. Вестник Башкирского государственного аграрного университета. 2022(2):119-24.
4. Котин Д.А, Чернов И.В, Мосин М.Е, Бутенко В.Н. Система электронного дифференциала полноприводного электромобиля // Интеллектуальный потенциал Сибири 2021 (pp. 321-324).
5. Лисов А.А., Кулганатов А.З., Панишев С.А.: Акустическое обнаружение транспортных средств аварийных служб с использованием сверхточных нейронных сетей // Инновационные транспортные системы и технологии. 2023. Т. 9. №1. С. 95-107. doi: 10.17816/transsyst20239195-107
6. Мосин М. Е., Бутенко В. Н., Чернов И. В. Система электронного дифференциала электромобиля с двумя приводными асинхронными двигателями // Современная наука: исследования и разработки. – 2020. – С. 90-93.
7. Попов Н.С. Повышение энергетической эффективности системы тягового электропривода безрельсового транспортного средства: диссертация кандидата технических наук: 05.09.03 / Попов Никита Сергеевич; [Место защиты: Новосибирский государственный технический университет]. - Новосибирск, 2022. - 150 с.: ил.
8. Aggarwal A. Electronic differential in electric vehicles. International Journal of Scientific & Engineering Research. 2013 Nov;4(11):1322-6..

9. Alrejgal A, Ksaibati K. Impact of combined alignments and adverse weather conditions on vehicle skidding. *Journal of traffic and transportation engineering (English edition)*. 2023 Feb 1;10(1):116-31.
10. Avilov O, Rimbert S, Popov A, Bougrain L.: Deep learning techniques to improve intraoperative awareness detection from electroencephalographic signals // 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC). 2020 P. 142-145.
11. Azam S, Munir F, Rafique MA, Sheri AM, Hussain MI, Jeon M. N 2 C: neural network controller design using behavioral cloning. *IEEE Transactions on Intelligent Transportation Systems*. 2021 Jan 1;22(7):4744-56.
12. Berkenkamp F, Turchetta M, Schoellig A, Krause A. Safe model-based reinforcement learning with stability guarantees. *Advances in neural information processing systems*. 2017;30.
13. Bishop CM. *Neural networks for pattern recognition*. Oxford university press; 1995 Nov 23.
14. Blalock D, Gonzalez Ortiz JJ, Frankle J, Guttag J. What is the state of neural network pruning?. *Proceedings of machine learning and systems*. 2020 Mar 15;2:129-46.
15. Bouguenna IF, Azaiz A, Tahour A, Larbaoui A. Robust neuro-fuzzy sliding mode control with extended state observer for an electric drive system. *Energy*. 2019 Feb 15;169:1054-63.
16. Bulat A, Kossaifi J, Tzimiropoulos G, Pantic M. Matrix and tensor decompositions for training binary neural networks. *arXiv preprint arXiv:1904.07852*. 2019 Apr 16.
17. Burgueno L, Cabot J, Li S, Gérard S. A generic LSTM neural network architecture to infer heterogeneous model transformations. *Software and Systems Modeling*. 2022 Feb;21(1):139-56.
18. Burgueño L, Cabot J, Gérard S. An LSTM-based neural network architecture for model transformations. In *2019 ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems (MODELS) 2019 Sep 15 (pp. 294-299)*. IEEE.

19. Capra M, Bussolino B, Marchisio A, Masera G, Martina M, Shafique M. Hardware and software optimizations for accelerating deep neural networks: Survey of current trends, challenges, and the road ahead. *IEEE Access*. 2020 Nov 24;8:225134-80.
20. Carlini N, Wagner D. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP) 2017 May 22 (pp. 39-57)*. Ieee.
21. Caruana R, Lou Y, Gehrke J, Koch P, Sturm M, Elhadad N. Intelligent models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining 2015 Aug 10 (pp. 1721-1730)*.
22. Chen C, Zhang Y, Gao Y. Learning how to self-learn: Enhancing self-training using neural reinforcement learning. In *2018 International Conference on Asian Language Processing (IALP) 2018 Nov 15 (pp. 25-30)*. IEEE.
23. Chen L, Li Z, Yang J, Song Y. Lateral stability control of four-wheel-drive electric vehicle based on coordinated control of torque distribution and ESP differential braking. In *Actuators 2021 Jun 18 (Vol. 10, No. 6, p. 135)*. MDPI.
24. Chen L, Zhu F, Zhang M, Huo Y, Yin C, Peng H. Design and analysis of an electrical variable transmission for a series-parallel hybrid electric vehicle. *IEEE Transactions on vehicular technology*. 2011 Apr 5;60(5):2354-63.
25. Chen S, Zhou J, Sun W, Huang L. Joint matrix decomposition for deep convolutional neural networks compression. *Neurocomputing*. 2023 Jan 7;516:11-26.
26. Cheng H, Zhang M, Shi JQ. A survey on deep neural network pruning: Taxonomy, comparison, analysis, and recommendations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2024 Aug 21.
27. Cheng Y, Wang D, Zhou P, Zhang T. Model compression and acceleration for deep neural networks: The principles, progress, and challenges. *IEEE Signal Processing Magazine*. 2018 Jan 10;35(1):126-36.
28. Cui H, Chen Q, Qi X, Wang H. Electric vehicle differential system based on co-simulation of Carsim/Simulink. In *2016 IEEE 11th Conference on Industrial Electronics and Applications (ICIEA) 2016 Jun 5 (pp. 1963-1966)*. IEEE.

29. Da Lio M, Donà R, Papini GP, Biral F, Svensson H. A mental simulation approach for learning neural-network predictive control (in self-driving cars). *IEEE Access*. 2020 Oct 21;8:192041-64.
30. David J, Brom P, Starý F, Bradáč J, Dynybyl V. Application of artificial neural networks to streamline the process of adaptive cruise control. *Sustainability*. 2021 Apr 20;13(8):4572.
31. Davidović J, Smailović E, Marković N, Antić B. The influence of weather conditions on road safety. *Journal of Road and Traffic Engineering*. 2017;63(4):13-20.
32. Deng L, Li G, Han S, Shi L, Xie Y. Model compression and hardware acceleration for neural networks: A comprehensive survey. *Proceedings of the IEEE*. 2020 Mar 20;108(4):485-532.
33. Devarakonda A., Naumov M., Garland M. AdaBatch: Adaptive Batch Sizes for Training Deep Neural Networks // *arXiv: Learning*. 2017. doi.org/10.48550/arXiv.1712.02029
34. Diamos G., Sengupta S., Catanzaro B., Chrzanowski, M., Coates A., Elsen, E., Engel J., Hannun A., Satheesh S.: Persistent RNNs: Stashing RecurrentWeights on-Chip // *International Conference on Machine Learning*. 2016. P. 2024–2033.
35. Dogo EM, Afolabi OJ, Nwulu NI, Twala B, Aigbavboa CO. A comparative analysis of gradient descent-based optimization algorithms on convolutional neural networks. In 2018 international conference on computational techniques, electronics and mechanical systems (CTEMS) 2018 Dec 21 (pp. 92-99). *IEEE*.
36. Dozat T. Incorporating nesterov momentum into adam, in *Int. Conference on Learning Representations, Vol. 1* (San Juan, Puerto Rico, 2016).
37. Duchi J, Hazan E, Singer Y. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*. 2011 Jul 1;12(7).
38. Emadi A, Lee YJ, Rajashekara K. Power electronics and motor drives in electric, hybrid electric, and plug-in hybrid electric vehicles. *IEEE Transactions on industrial electronics*. 2008 May 28;55(6):2237-45.

39. Fang S, Song J, Song H, Tai Y, Li F, Nguyen TS. Design and control of a novel two-speed uninterrupted mechanical transmission for electric vehicles. *Mechanical Systems and Signal Processing*. 2016 Jun 15;75:473-93.
40. Farag W, Saleh Z. Behavior cloning for autonomous driving using convolutional neural networks. In *2018 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT) 2018 Nov 18* (pp. 1-7). IEEE.
41. Favaro F, Eurich S, Nader N. Autonomous vehicles' disengagements: Trends, triggers, and regulatory limitations. *Accident Analysis & Prevention*. 2018 Jan 1;110:136-48.
42. Gama J, Zliobaitė I, Bifet A, Pechenizkiy M, Bouchachia A. A survey on concept drift adaptation. *ACM computing surveys (CSUR)*. 2014 Mar 1;46(4):1-37.
43. Gavgani AM, Bingham T, Sorniotti A, Doherty J, Cavallino C, Fracchia M. A parallel hybrid electric drivetrain layout with torque-fill capability. *SAE International Journal of Passenger Cars-Mechanical Systems*. 2015 Jul 1;8(2015-01-9108):767-78.
44. Gholami A, Kim S, Dong Z, Yao Z, Mahoney MW, Keutzer K. A survey of quantization methods for efficient neural network inference. In *Low-Power Computer Vision 2022 Feb 22* (pp. 291-326). Chapman and Hall/CRC.
45. Gonyaev AG. The visual solution layout of the twin-seat electric car. *Молодежь и современные информационные технологии: сборник трудов XIII Международной научно-практической конференции студентов, аспирантов и молодых ученых, г. Томск, 9-13 ноября 2015 г. Т. 2.—Томск, 2016. 2016 (Vol. 2, pp. 210-211). Изд-во ТПУ.*
46. Goodfellow I, Bengio Y, Courville A. *Deep learning*. MIT press; 2016 Nov 10.
47. Goodman B, Flaxman S. European Union regulations on algorithmic decision-making and a “right to explanation”. *AI magazine*. 2017 Oct 2;38(3):50-7.
48. Goyal P., Dollar P., Girshick R., Noordhuis P., Wesolowski L., Kyrola A., Tulloch A., Jia Y., He K. Accurate, Large Minibatch SGD: Training ImageNet in 1 Hour // *arXiv: Computer Vision and Pattern Recognition*. 2017. doi.org/10.48550/arXiv.1706.02677

49. Guo Y. A survey on methods and theories of quantized neural networks. arXiv preprint arXiv:1808.04752. 2018 Aug 13.
50. Gupta A, Johnson J, Fei-Fei L, Savarese S, Alahi A. Social gan: Socially acceptable trajectories with generative adversarial networks. In Proceedings of the IEEE conference on computer vision and pattern recognition 2018 (pp. 2255-2264).
51. Haddoun A, Benbouzid ME, Diallo D, Abdessemed R, Ghouili J, Srairi K. Analysis, modeling and neural network traction control of an electric vehicle without differential gears. In 2007 IEEE International Electric Machines & Drives Conference 2007 May 3 (Vol. 1, pp. 854-859). IEEE.
52. Haddoun A, Benbouzid ME, Diallo D, Abdessemed R, Ghouili J, Srairi K. Modeling, analysis, and neural network control of an EV electrical differential. IEEE Transactions on industrial electronics. 2008 May 28;55(6):2286-94.
53. Haddoun A, Khoucha F, Benbouzid ME, Diallo D, Abdessemed R, Srairi K. SDTC neural network traction control of an electric vehicle without differential gears. In 2007 IEEE Vehicle Power and Propulsion Conference 2007 Sep 9 (pp. 259-266). IEEE.
54. He Y, Zhou X, Liu Z, Ran M. Layout optimization design of electric vehicle charging station based on urban parking lot. In Green Intelligent Transportation Systems: Proceedings of the 7th International Conference on Green Intelligent Transportation System and Safety 7 2018 (pp. 399-407). Springer Singapore.
55. Hochreiter S, Schmidhuber J. Long short-term memory. Neural computation. 1997 Nov 15;9(8):1735-80.
56. Hussain MT, Sulaiman NB, Hussain MS, Jabir M. Optimal Management strategies to solve issues of grid having Electric Vehicles (EV): A review. Journal of Energy Storage. 2021 Jan 1;33:102-114.
57. Jarrahi MH. Artificial intelligence and the future of work: Human-AI symbiosis in organizational decision making. Business horizons. 2018 Jul 1;61(4):577-86.
58. Jastrzebski S., Kenton Z., Arpit D., Ballas N., Fischer, A., Bengio Y., Storkey A.: Finding Flatter Minima with SGD // International Conference on Learning Representations. 2018.

59. Ji X, He X, Lv C, Liu Y, Wu J. A vehicle stability control strategy with adaptive neural network sliding mode theory based on system uncertainty approximation. *Vehicle System Dynamics*. 2018 Jun 3;56(6):923-46.
60. Kahveci H, Okumus HI, Ekici M. An electronic differential system using fuzzy logic speed controlled in-wheel brushless DC motors. In *4th International Conference on Power Engineering, Energy and Electrical Drives 2013* May 13 (pp. 881-885). IEEE.
61. Kalla D, Smith N, Samaah F, Kuraku S. Study and analysis of chat GPT and its impact on different fields of study. *International journal of innovative science and research technology*. 2023 Mar 1;8(3).
62. Kamal MA, Mukai M, Murata J, Kawabe T. Ecological vehicle control on roads with up-down slopes. *IEEE Transactions on Intelligent Transportation Systems*. 2011 Feb 28;12(3):783-94.
63. Katz G, Barrett C, Dill DL, Julian K, Kochenderfer MJ. Reluplex: An efficient SMT solver for verifying deep neural networks. In *Computer Aided Verification: 29th International Conference, CAV 2017, Heidelberg, Germany, July 24-28, 2017, Proceedings, Part I* 30 2017 (pp. 97-117). Springer International Publishing.
64. Keskar N. S., Mudigere D., Nocedal J., Smelyanskiy M., Tang P. T. P.: On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima // *International Conference on Learning Representations*. 2017. doi.org/10.48550/arXiv.1609.04836
65. Khan J, Wei JS, Ringner M, Saal LH, Ladanyi M, Westermann F, Berthold F, Schwab M, Antonescu CR, Peterson C, Meltzer PS. Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. *Nature medicine*. 2001 Jun;7(6):673-9.
66. Kim N, Bae J, Kim C, Park S, Sohn HG. Object distance estimation using a single image taken from a moving rolling shutter camera. *Sensors*. 2020 Jul 10;20(14):3860.
67. Kim T, Oh J, Kim N, Cho S, Yun SY. Comparing kullback-leibler divergence and mean squared error loss in knowledge distillation. *arXiv preprint arXiv:2105.08919*. 2021 May 19.

68. Kingma DP, Ba J. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980. 2014 Dec 22.
69. Kurbiel T, Khaleghian S. Training of deep neural networks based on distance measures using RMSProp. arXiv preprint arXiv:1708.01911. 2017 Aug 6.
70. Lapuschkin S, Wäldchen S, Binder A, Montavon G, Samek W, Müller KR. Unmasking Clever Hans predictors and assessing what machines really learn. *Nature communications*. 2019 Mar 11;10(1):1096.
71. Laurent T, von Brecht J. A recurrent neural network without chaos. arXiv preprint arXiv:1612.06212. 2016 Dec 19.
72. LeCun Y, Bengio Y, Hinton G. Deep learning. *nature*. 2015 May 28;521(7553):436-44.
73. LeCun Y, Bottou L, Orr GB, Müller KR. Efficient backprop. In *Neural networks: Tricks of the trade 2002* Mar 28 (pp. 9-50). Berlin, Heidelberg: Springer Berlin Heidelberg.
74. LeCun Y. A., Bottou, L., Orr G. B., Muller K.: *Neural Networks: Tricks of the Trade* // Springer, Berlin, GER. 2012.
75. Lee CP, Lin CJ. A study on L2-loss (squared hinge-loss) multiclass SVM. *Neural computation*. 2013 May;25(5):1302-23.
76. Lee JS, Ryoo YJ, Lim YC, Freere P, Kim TG, Son SJ, Kim ES. A neural network model of electric differential system for electric vehicle. In *2000 26th Annual Conference of the IEEE Industrial Electronics Society. IECON 2000. 2000 IEEE International Conference on Industrial Electronics, Control and Instrumentation. 21st Century Technologies 2000* Oct 22 (Vol. 1, pp. 83-88). IEEE.
77. Li M, Zhang T, Chen Y, Smola AJ.: Efficient mini-batch training for stochastic optimization // *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2014 P. 661-670. doi.org/10.1145/2623330.2623612
78. Li SE. Deep reinforcement learning. In *Reinforcement learning for sequential decision and optimal control 2023* Apr 6 (pp. 365-402). Singapore: Springer Nature Singapore.

79. Li Z, Li H, Meng L. Model compression for deep neural networks: A survey. *Computers*. 2023 Mar 12;12(3):60.
80. Lin C, Xu Z, Zhang R. A Yaw Stability Control Algorithm for Four-Wheel Independently Actuated Electric Ground Vehicles considering Control Boundaries. *Mathematical Problems in Engineering*. 2015;2015(1):256715.
81. Lin TY, Goyal P, Girshick R, He K, Dollár P. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision 2017* (pp. 2980-2988).
82. Lin Z., Courbariaux M., Memisevic R., Bengio Y.: Neural networks with Few Multiplications // *Proceedings of the 32d International Conference on Machine Learning, ICML*. 2016. Vol. 16. P. 561–568. doi.org/10.48550/arXiv.1510.03009
83. Lisov AA., Vozmilov AG., Urmanov VG., Panishev SA. The Impact of Batch Size on the Quality of Training of Neural Networks. *Modern Information Technologies and IT-Education*. 2023;19(2):324-332.
84. Lu J. Matrix decomposition and applications. *arXiv preprint arXiv:2201.00145*. 2022 Jan 1.
85. Mahadika P, Subiantoro A, Kusumoputro B. Neural network predictive control approach design for adaptive cruise control. *Int. J. Technol*. 2020 Dec 15;11:1451.
86. Mallozzi P, Pelliccione P, Knauss A, Berger C, Mohammadiha N. Autonomous vehicles: state of the art, future trends, and challenges. *Automotive systems and software engineering: State of the art and future trends*. 2019:347-67.
87. Meier T, Rinderknecht S, Fietzek R. Electric power train configurations with appropriate transmission systems. *SAE Technical Paper*; 2011 Apr 12.
88. Mester V, Gillon F, Brochet P. Optimal design of electric vehicles drive train. *COMPEL-The international journal for computation and mathematics in electrical and electronic engineering*. 2005 Sep 1;24(3):967-83.
89. Mihai PA, VIOREL A. A Possible Layout of a Serial Hybrid Electric Car Powertrain. *Advances in Electrical and Computer Engineering*. 2001 Apr 2;1(1):11-4.

90. Mihet-Popa L, Saponara S. Toward green vehicles digitalization for the next generation of connected and electrified transport systems. *Energies*. 2018 Nov 12;11(11):3124.
91. Nam K, Oh S, Fujimoto H, Hori Y. Robust yaw stability control for electric vehicles based on active front steering control through a steer-by-wire system. *International Journal of Automotive Technology*. 2012 Dec;13:1169-76.
92. Nguyen A, Yosinski J, Clune J. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE conference on computer vision and pattern recognition 2015* (pp. 427-436).
93. Nguyen DH, Widrow B. Neural networks for self-learning control systems. *IEEE Control systems magazine*. 1990 Apr;10(3):18-23.
94. Papadopoulos G, Edwards PJ, Murray AF. Confidence estimation methods for neural networks: A practical comparison. *IEEE transactions on neural networks*. 2001 Nov;12(6):1278-87.
95. Perez-Pinal FJ, Cervantes I, Emadi A. Stability of an electric differential for traction applications. *IEEE Transactions on Vehicular Technology*. 2009 Jan 27;58(7):3224-33.
96. Powell M.J. *Radial Basis Functions for Multivariable Interpolation: A Review* // Clarendon Press. 1987. P. 143-167.
97. Qi J, Du J, Siniscalchi SM, Ma X, Lee CH. On mean absolute error for deep neural network based vector-to-vector regression. *IEEE Signal Processing Letters*. 2020 Aug 14;27:1485-9.
98. Radunovic B, Le Boudec JY. A unified framework for max-min and min-max fairness with applications. *IEEE/ACM Transactions on networking*. 2007 Oct 15;15(5):1073-83.
99. Ranjan S, Senthamilarasu S. *Applied Deep Learning and Computer Vision for Self-Driving Cars: Build autonomous vehicles using deep neural networks and behavior-cloning techniques*. Packt Publishing Ltd; 2020 Aug 14.

100. Razaviyayn M, Huang T, Lu S, Nouiehed M, Sanjabi M, Hong M. Nonconvex min-max optimization: Applications, challenges, and recent theoretical advances. *IEEE Signal Processing Magazine*. 2020 Sep 3;37(5):55-66.
101. Reddi SJ, Kale S, Kumar S. On the convergence of adam and beyond. *arXiv preprint arXiv:1904.09237*. 2019 Apr 19.
102. Ren Q, Crolla DA, Morris A. Effect of transmission design on electric vehicle (EV) performance. In *2009 IEEE vehicle power and propulsion conference 2009 Sep 7* (pp. 1260-1265). IEEE.
103. Rolnick D, Tegmark M. The power of deeper networks for expressing natural functions. *arXiv preprint arXiv:1705.05502*. 2017 May 16.
104. Rosenblatt F. *Principles of neurodynamics. perceptrons and the theory of brain mechanisms // Cornell Aeronautical Lab Inc Buffalo NY*. 1961.
105. Ruby U, Yendapalli V. Binary cross entropy with deep learning technique for image classification. *Int. J. Adv. Trends Comput. Sci. Eng*. 2020 Jul;9(10).
106. Ruder S. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*. 2016 Sep 15.
107. Sakai SI, Sado H, Hori Y. Motion control in an electric vehicle with four independently driven in-wheel motors. *IEEE/ASME Transactions on mechatronics*. 1999 Mar;4(1):9-16.
108. Samek W, Montavon G, Vedaldi A, Hansen LK, Müller KR, editors. *Explainable AI: interpreting, explaining and visualizing deep learning*. Springer Nature; 2019 Sep 10.
109. Schaul T, Antonoglou I, Silver D. Unit tests for stochastic optimization. *arXiv preprint arXiv:1312.6055*. 2013 Dec 20.
110. Schmidhuber J. Deep learning in neural networks: An overview. *Neural networks*. 2015 Jan 1;61:85-117.
111. Schumacher S, Schmid S, Wieser P, Stetter R, Till M. Design, Simulation and Optimization of an Electrical Drive-Train. *Vehicles*. 2021 Jul 27;3(3):390-405.

112. Schütt KT, Arbabzadah F, Chmiela S, Müller KR, Tkatchenko A. Quantum-chemical insights from deep tensor neural networks. *Nature communications*. 2017 Jan 9;8(1):13890.
113. Serrano-Pérez JD, Fernández-Anaya G, Carrillo-Moreno S, Yu W. New results for prediction of chaotic systems using deep recurrent neural networks. *Neural Processing Letters*. 2021 Apr;53:1579-96.
114. Sharma S, Tewolde G, Kwon J. Behavioral cloning for lateral motion control of autonomous vehicles using deep learning. In *2018 IEEE International Conference on Electro/Information Technology (EIT) 2018 May 3* (pp. 0228-0233). IEEE.
115. Smith S. L., Kindermans P., Le Q. V.: Don't Decay the Learning Rate, Increase the Batch Size // *International Conference on Learning Representations*. 2018. doi.org/10.48550/arXiv.1711.00489
116. Su D, Zhang H, Chen H, Yi J, Chen PY, Gao Y. Is robustness the cost of accuracy?-a comprehensive study on the robustness of 18 deep image classification models. In *Proceedings of the European conference on computer vision (ECCV) 2018* (pp. 631-648).
117. Sumanth U, Punn NS, Sonbhadra SK, Agarwal S. Enhanced behavioral cloning-based self-driving car using transfer learning. In *Data Management, Analytics and Innovation: Proceedings of ICDMAI 2021, Volume 2 2021 Sep 20* (pp. 185-198). Singapore: Springer Singapore.
118. Sutskever I, Martens J, Dahl G, Hinton G. On the importance of initialization and momentum in deep learning. In *International conference on machine learning 2013 May 26* (pp. 1139-1147). PMLR.
119. Szegedy C, Toshev A, Erhan D. Deep neural networks for object detection. *Advances in neural information processing systems*. 2013;26.
120. Tabbache B, Kheloui A, Hanini N. An electric differential system for a two-wheel mobile plat-form using direct torque control with adaptive flux and speed observers. In *2008 International Symposium on Power Electronics, Electrical Drives, Automation and Motion 2008 Jun 11* (pp. 550-556). IEEE.

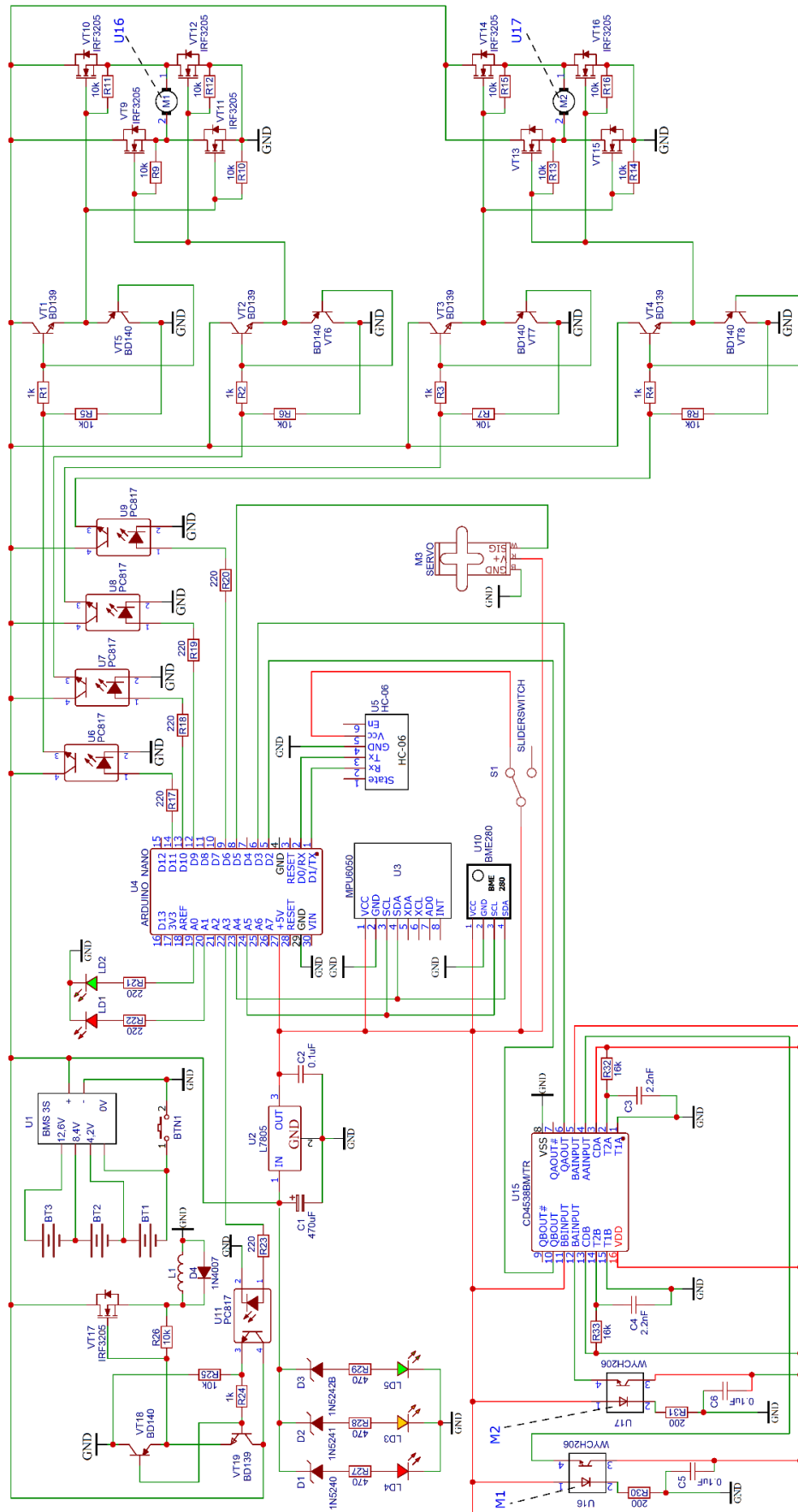
121. Teng TH, Tan AH, Zurada JM. Self-organizing neural networks integrating domain knowledge and reinforcement learning. *IEEE transactions on neural networks and learning systems*. 2014 Jun 20;26(5):889-902.
122. Van den Oord A., Dieleman S., Schrauwen B. Deep content-based music recommendation // *Advances in neural information processing systems*. 2013.
123. Vodovozov V, Aksjonov A, Petlenkov E, Raud Z. Neural network-based model reference control of braking electric vehicles. *Energies*. 2021 Apr 22;14(9):2373.
124. Vozmilov A., Andreev L., Lisov A.: Development of an algorithm for the program to recognize defects on the surface of hot-rolled metal // 2022 International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM). – IEEE. 2022. – P. 1004-1008.
125. Vozmilov A., Urmanov V., Lisov A.: Using computer vision to recognize defects on the surface of hot-rolled steel // 2022 International Ural Conference on Electrical Power Engineering (UralCon). – IEEE. 2022. – P. 21-25. DOI: 10.1109/UralCon54942.2022.9906737
126. Wang H, Qin C, Bai Y, Zhang Y, Fu Y. Recent advances on neural network pruning at initialization. *arXiv preprint arXiv:2103.06460*. 2021 Mar 11.
127. Wang M, Yu H, Dong G, Huang M. Dual-mode adaptive cruise control strategy based on model predictive control and neural network for pure electric vehicles. In 2019 5th international conference on transportation information and safety (ICTIS) 2019 Jul 14 (pp. 1220-1225). IEEE.
128. Warnecke A, Arp D, Wressnegger C, Rieck K. Evaluating explanation methods for deep learning in security. In 2020 IEEE european symposium on security and privacy (EuroS&P) 2020 Sep 7 (pp. 158-174). IEEE.
129. Weng O. Neural network quantization for efficient inference: A survey. *arXiv preprint arXiv:2112.06126*. 2021 Dec 8.
130. Wilson DR, Martinez TR.: The general inefficiency of batch training for gradient descent learning // *Neural networks*. 2003.Vol. 16. issue 10. P.1429-1451. doi.org/10.1016/S0893-6080(03)00138-2

131. Wittmann J. Electrification and digitalization as disruptive trends: new perspectives for the automotive industry?. *Phantom ex machina: Digital disruption's role in business model transformation*. 2017:137-59.
132. Xiaofei C, Pinga L, Mingliang Y, Lei S, Liquan L. Electronic differential control strategy for four-wheel wheel drive vehicle based on slip ratio. *Journal of Chongqing University of Technology (Natural Science)*. 2019;33(03):51-8.
133. Yildirim M, Catalbas MC, Gulden A, Kurum H. Computation of the speed of four in-wheel motors of an electric vehicle using a radial basis neural network. *Engineering, Technology & Applied Science Research*. 2016 Dec 18;6(6):1288-93.
134. Yildirim M, Kurum H. Electronic differential system for an electric vehicle with four in-wheel PMSM. In *2020 IEEE 91st vehicular technology conference (VTC2020-Spring)* 2020 May 25 (pp. 1-5). IEEE.
135. Yildirim M, Öksüztepe E, Tanyeri B, Kürüm H. Electronic differential system for an electric vehicle with in-wheel motor. *2015 9th International Conference on Electrical and Electronics Engineering (ELECO)* 2015 Nov 26 (pp. 1048-1052). IEEE.
136. Yildirim M, Öksüztepe E, Tanyeri B, Kürüm H. Electronic differential system for an electric vehicle with in-wheel motor. In *2015 9th International Conference on Electrical and Electronics Engineering (ELECO)* 2015 Nov 26 (pp. 1048-1052). IEEE.
137. Yu H, Castelli-Dezza F, Cheli F. Optimal powertrain design and control of a 2-IWD electric race car. *2017 International Conference of Electrical and Electronic Technologies for Automotive* 2017 Jun 15 (pp. 1-7). IEEE.
138. Zeiler MD. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*. 2012 Dec 22.
139. Zerroug A, Terrissa L, Faure A. Chaotic dynamical behavior of recurrent neural network. *Annu. Rev. Chaos Theory Bifurc. Dyn. Syst.* 2013;4:55-66.
140. Zhang N, Wang J, Li Z, Li S, Ding H. Multi-agent-based coordinated control of ABS and AFS for distributed drive electric vehicles. *Energies*. 2022 Mar 6;15(5):1919.
141. Zhang W, Liu Z, Chen Q. Electronic differential system based on adaptive SMC combined with QP for 4WID electric vehicles. *World Electric Vehicle Journal*. 2021 Aug 20;12(3):126.

142. Zhang W, Liu Z, Chen Q. Electronic differential system based on adaptive SMC combined with QP for 4WID electric vehicles. *World Electric Vehicle Journal*. 2021 Aug 20;12(3):126.

143. Zhu W, Anderson B, Zhu S, Wang Y. A computer vision-based system for stride length estimation using a mobile phone camera. In *Proceedings of the 18th International ACM SIGACCESS Conference on Computers and Accessibility* 2016 Oct 23 (pp. 121-130).

Приложение А. Принципиальная схема разработанной физической модели электромобиля и использованные компоненты.

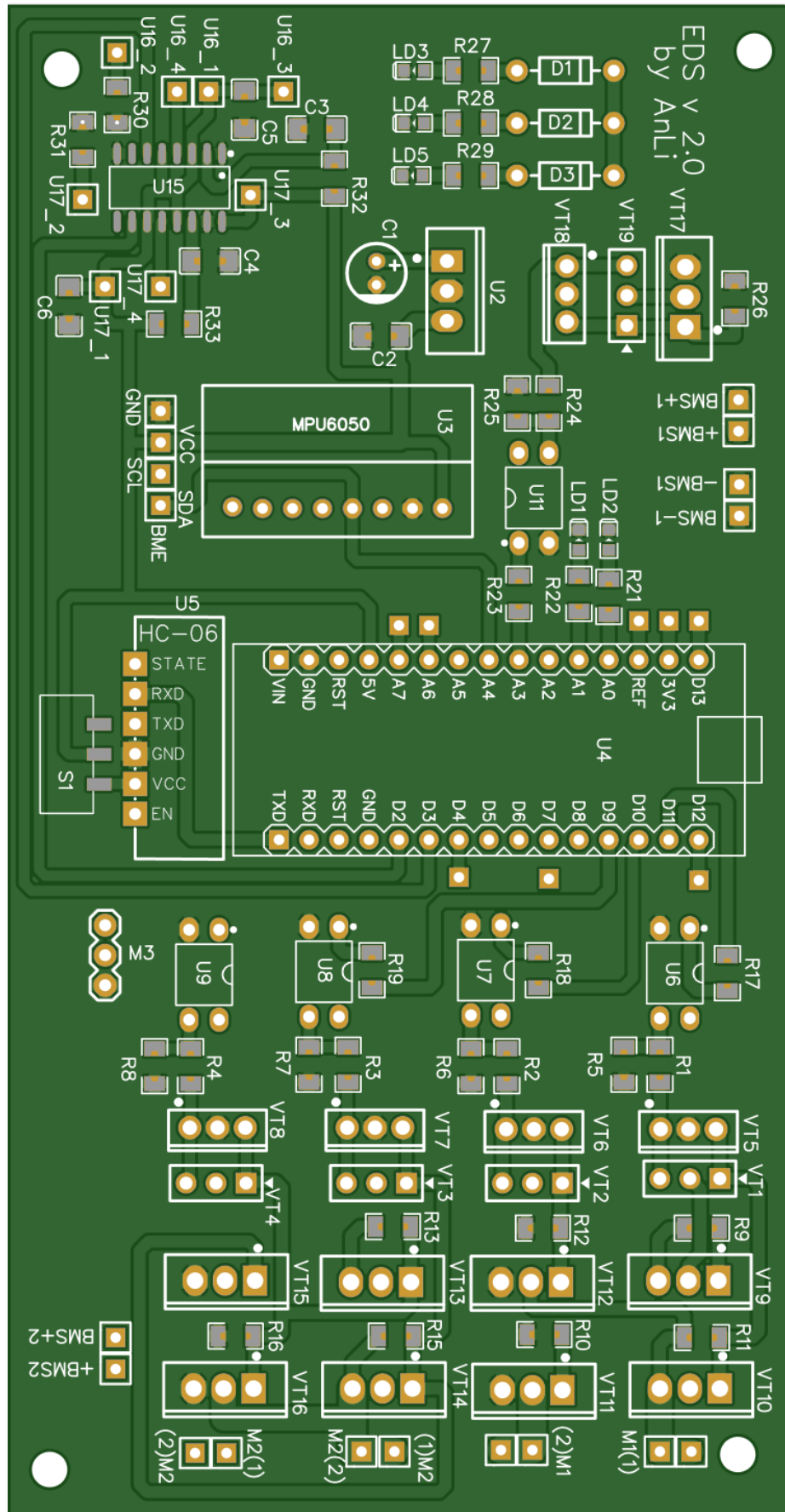


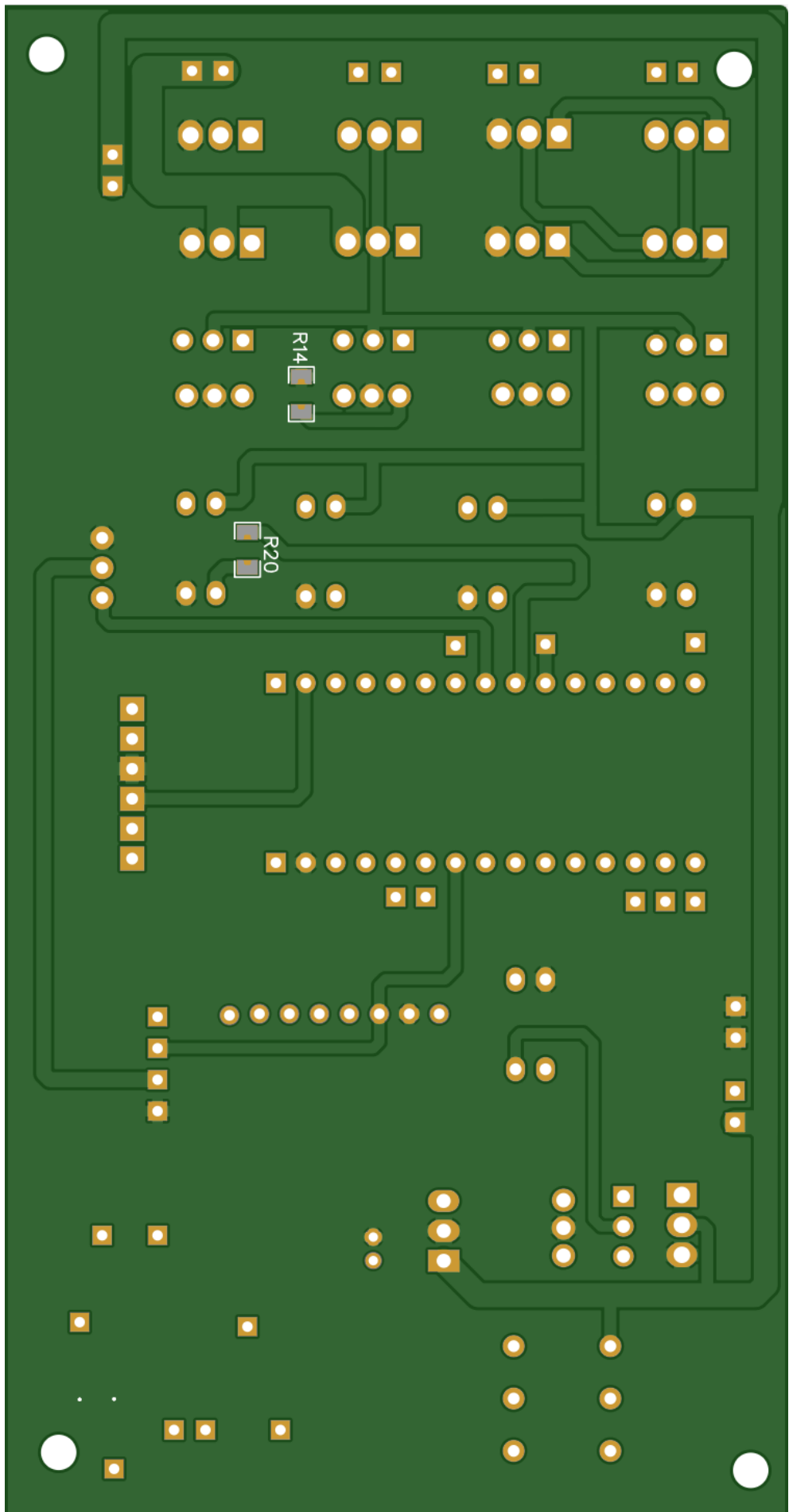
Использованные компоненты для разработки модели электромобиля.

№	Компонент	Название, номинал	Условное обозначение	Количество
1	Конденсатор	470uF	C1	1
2	Конденсатор	0.1uF	C2, C5, C6	1
3	Конденсатор	2.2nF	C3, C4	2
4	Светодиоды	LED 1206	LD1, LD2	2
5	Щёточный двигатель	JGA25-370, 12 В, 1200 об/мин	M1, M2	2
6	Серводвигатель рулевых тяг	HiTec HS-5485HB	M3	1
7	Резистор	1 кОм	R1–R4, R24	5
8	Резистор	10 кОм	R5–R16, R25, R26	14
9	Резистор	220 Ом	R17 – R23, R30, R31	9
10	Резистор	16 кОм	R32, R33	2
11	Балансир для АКБ	BMS_3S	U1	1
12	Стабилизатор напряжения	L7805	U2	1
13	Гироскоп + акселерометр	MPU6050	U3	1
14	Микроконтроллер	ARDUINO_NANO	U4	1
15	Bluetooth-модуль	HC-06	U5	1
16	АКБ	LiitoKala HG2, 3000 мАч	BT1–BT3	3
17	Оптопара	PC817	U6–U9, U11	5
18	Датчик влажности, давления и температуры	BME280	U10	1
19	Транзистор, NPN	BD139	VT1–VT4, VT19	5
20	Транзистор, PNP	BD140	VT5–VT8, VT18	5
21	Полевой транзистор, n- канальный	IRF3205	VT9–VT17	9
22	Переключатель-слайдер	SLIDERSWITCH	S1	1

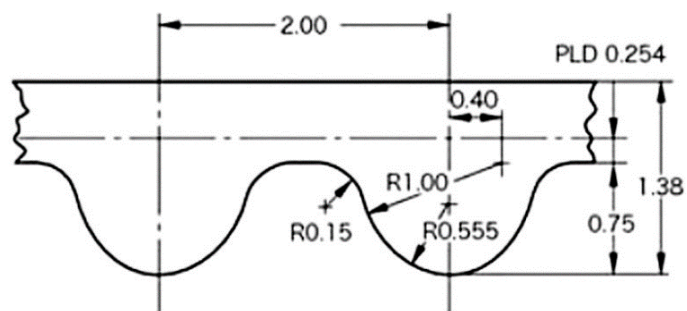
23	Оптический датчик	WYC H206	U16, U17	2
24	Соленоид	TAU-0537, 12B	L1	1
25	Кнопка тактовая	K2-3.6×6.1_SMD	BTN1	1
26	Двойной прецизионный моностабильный мультивибратор	CD4538BM/TR	U15	1

**Приложение Б. Печатная плата разработанной модели электромобиля с
электронной дифференциальной системой**

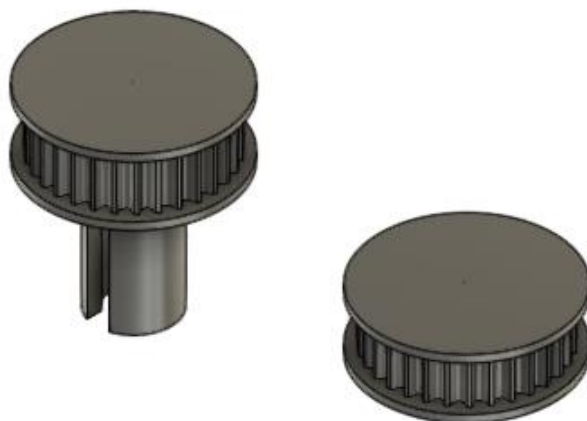




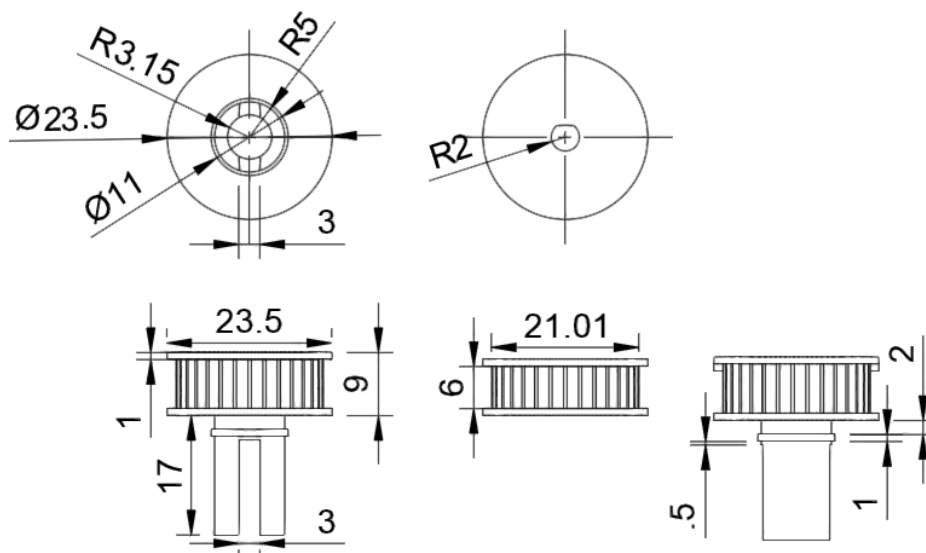
Приложение В. Внешний вид и размеры передаточного механизма для тягового привода модели электроавтомобиля



Параметры ремня для передачи вращательного момента



Внешний вид ведомой (слева) и ведущей (справа) шестерен



Геометрические размеры ведущих и ведомых шестерен

Приложение Г. ПО для обучения нейронной сети на подготовленных данных.

1 ячейка. Подключим Гугл.Диск для сохранения и загрузки моделей

```
from google.colab import drive
drive.mount('/content/drive')
! pip install pyyaml h5py # Требуется для сохранения и загрузки моделей в
HDF5-формате
```

2 ячейка. Настройки

```
number_of_interations = 11 #Номер итерации для сохранения и загрузки
готовой модели
split_percent = 0.9 # Процент данных для обучения

BATCH_SIZE = 16 # Размер пакета для обучения
EPOCHS = 65 # Количество эпох обучения
NN_optimizer = 'adamax' # Оптимизатор для модели НС
NN_loss = 'mse' # Функция потерь
csv_file =
'https://raw.githubusercontent.com/AnLiMan/Electronics_Differential_System/
main/Datasets/Dataset_5.csv' # Ссылка на датасет
save_path = "/content/drive/My Drive/Files for colab/NN for diss/NN_model_"
+ str(number_of_interations) + ".h5" # Путь для сохранения и загрузки
моделей нейронки с гугл диска
print('Текущий номер для сохранения модели, чекпоинтов, а также загрузки
существующей модели с облака: ', number_of_interations)
```

3 ячейка. Библиотеки

```
import pandas as pd
import numpy as np

import tensorflow as tf
from keras import layers
from tensorflow import keras
import math
from sklearn.metrics import mean_squared_error, accuracy_score
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
import random
import time
```

4 ячейка. Загрузим датасет, выведем некоторую информацию и совершим серию проверок

```
df = pd.read_csv(csv_file) # Прочтём датасет
df.head() # Выведем заголовки
```

5 ячейка

```
df.dtypes # Используемые типы данных
```

6 ячейка

```
print(df.info())
```

7 ячейка

```
print(df.shape)
```

8 ячейка. Проверим наличие отсутствующих значений.

```
print(df.isna().sum())
```

9 ячейка. Проверим на дубликаты

```
print(df.duplicated(keep = False).value_counts())  
print(df[df.duplicated(keep = False)])
```

10 ячейка. Выкинем дубликаты

```
df = df.drop_duplicates(keep = 'first').copy()  
print(df.shape)
```

11 ячейка. Сбросим индекс dataframe

```
df = df.reset_index(drop = True)
```

12 ячейка. Выделим колонки из датасета, которые нужно будет предсказывать в будущем (Y).

```
target_names = ['V1', 'V2']  
y = df[target_names]  
y.head()
```

13 ячейка. Выделим все независимые переменные из датасета, значений типа "строка" здесь быть не должно.

```
feature_names = ['AngleServo', 'V' 'ax', 'ay', 'az', 'gx', 'gy', 'gz',  
'temp', 'Deviation']  
X = df[feature_names]  
X.head()
```

14 ячейка. Проверим размерности

```
print(f"Размерность X, строки(общее количество значений), колонны  
(количество независимых переменных) = {X.shape}")  
print(f"Размерность Y, строки(общее количество значений), колонны  
(количество зависимых переменных) = {y.shape} ")
```

15 ячейка.

```
print("Значения X ")  
print(X)
```

16 ячейка.

```
print("Значения Y")  
print(y)
```

17 ячейка. Далее данные следует нормализовать, но т.к. это было сделано ранее, то этот этап можно пропустить и приступить к разделению данных на тренировочную и тестовые выборки

```
X_train, X_valid, y_train, y_valid = train_test_split(X, y, test_size = 1 -  
split_percent)
```

18 ячейка.

```
print("Тренировочные данные")  
print(X_train)
```

19 ячейка.

```
print("Выходные данные для тренировочных")  
print(y_train)
```

21 ячейка.

```
print("Данные для валидации (проверки)")
print(X_valid)
```

22 ячейка.

```
print("Выходные данные для валидации (проверки)")
print(y_valid)
```

23 ячейка. Модель нейросети

```
def get_basic_model():
    model = tf.keras.models.Sequential([
        tf.keras.layers.Flatten(),
        tf.keras.layers.Dense(32, activation='relu', input_shape = (10,)),
        tf.keras.layers.Dense(128, activation='relu'),
        tf.keras.layers.Dense(512, activation='relu'),
        tf.keras.layers.Dense(512, activation='relu'),
        tf.keras.layers.Dropout(0.35),
        tf.keras.layers.Dense(2)])

    model.compile(metrics = ['accuracy'], optimizer = NN_optimizer, loss =
NN_loss)
    return model
```

24 ячейка. Обучение нейросети и её сохранение

```
model = get_basic_model() #Получаем модель и компилируем её

#Обучаем
start_time = time.time()
history = model.fit(X_train, y_train,
                    batch_size = BATCH_SIZE,
                    epochs = EPOCHS,
                    verbose = 1,
                    validation_data = (X_valid, y_valid))

print("\nВремя обучения = ", (time.time() - start_time), ' c')
model.summary() #Выводим резюме
model.save(save_path) #Сохраняем
```

25 ячейка. Точность прогнозирования модели

```
plt.figure(figsize=(10, 5), dpi=90)
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('Точность прогнозирования модели')
plt.ylabel('Точность')
plt.xlabel('Эпоха')
plt.grid()
plt.legend(['train', 'test'], loc='upper left')
```

26 ячейка. Потери модели

```
plt.figure(figsize=(10, 5), dpi=90)
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('Потери модели')
```

```
plt.ylabel('Потери')
plt.grid()
plt.xlabel('Эпоха')
plt.legend(['train', 'test'], loc='upper right')
plt.show()
```

27 ячейка. Работа с загруженной моделью

```
Loading_model = tf.keras.models.load_model(save_path)
Loading_model.summary()
```

28 ячейка. Визуализация архитектуры модели

```
keras.utils.plot_model>Loading_model, 'Model.png')
```

29 ячейка.

```
keras.utils.plot_model>Loading_model, 'model_with_shape_info.png',
show_shapes=True)
```

30 ячейка. Проверка точности предсказания

```
data = [[ 1.0000, 0.9529, -0.0297, -0.0028, 0.6945, 0.0000, 0.0000,
0.0000, 0.8227, 0.0000]] # X_valid[0]
print(f"Предсказание = {Loading_model.predict(data)}")
print(f"Реальное значение = 0.804 0.740")
```

31 ячейка. Преобразование данных в массив numpy

```
test_arr = np.array(y_valid, ndmin=2)
predict_arr = np.array(X_valid, ndmin=2)
test_predict = Loading_model.predict(predict_arr)
```

32 ячейка. График тестовой выборки

```
def plot_result(testY, test_predict):
    plt.figure(figsize=(15, 6), dpi=90)
    plt.plot(testY)
    plt.plot(test_predict)
    plt.legend(['Реальное V1', 'Реальное V2', 'Предсказанное V1', 'Предсказанное
V2'])
    plt.xlabel('x')
    plt.ylabel('y')
    plt.title('Реальное и предсказанные значения')
    plt.grid()
plot_result(test_arr, test_predict)
```

33 ячейка. Прогон по тестовой выборке

```
for i in range(len(X_valid)):
    data = np.array(predict_arr[i], ndmin=2)
    y = Loading_model.predict(data)
    print(f" {i}-е Предсказание при вводных данных {data} = {y}")
    print(f"{i}-е Реальное значение = {test_arr[i]} ")
```

34 ячейка. Root Mean Square Error (RMSE) - Среднеквадратичное отклонение

```
def print_error(trainY, testY, train_predict, test_predict):
    # Ошибка предсказания
    train_rmse = math.sqrt(mean_squared_error(trainY, train_predict))
    test_rmse = math.sqrt(mean_squared_error(testY, test_predict))
```

```

# Печать значений
print('RMSE для тренировочной выборки: %.3f RMSE' % (train_rmse))
print('RMSE для тестовой выборки: %.3f RMSE' % (test_rmse))

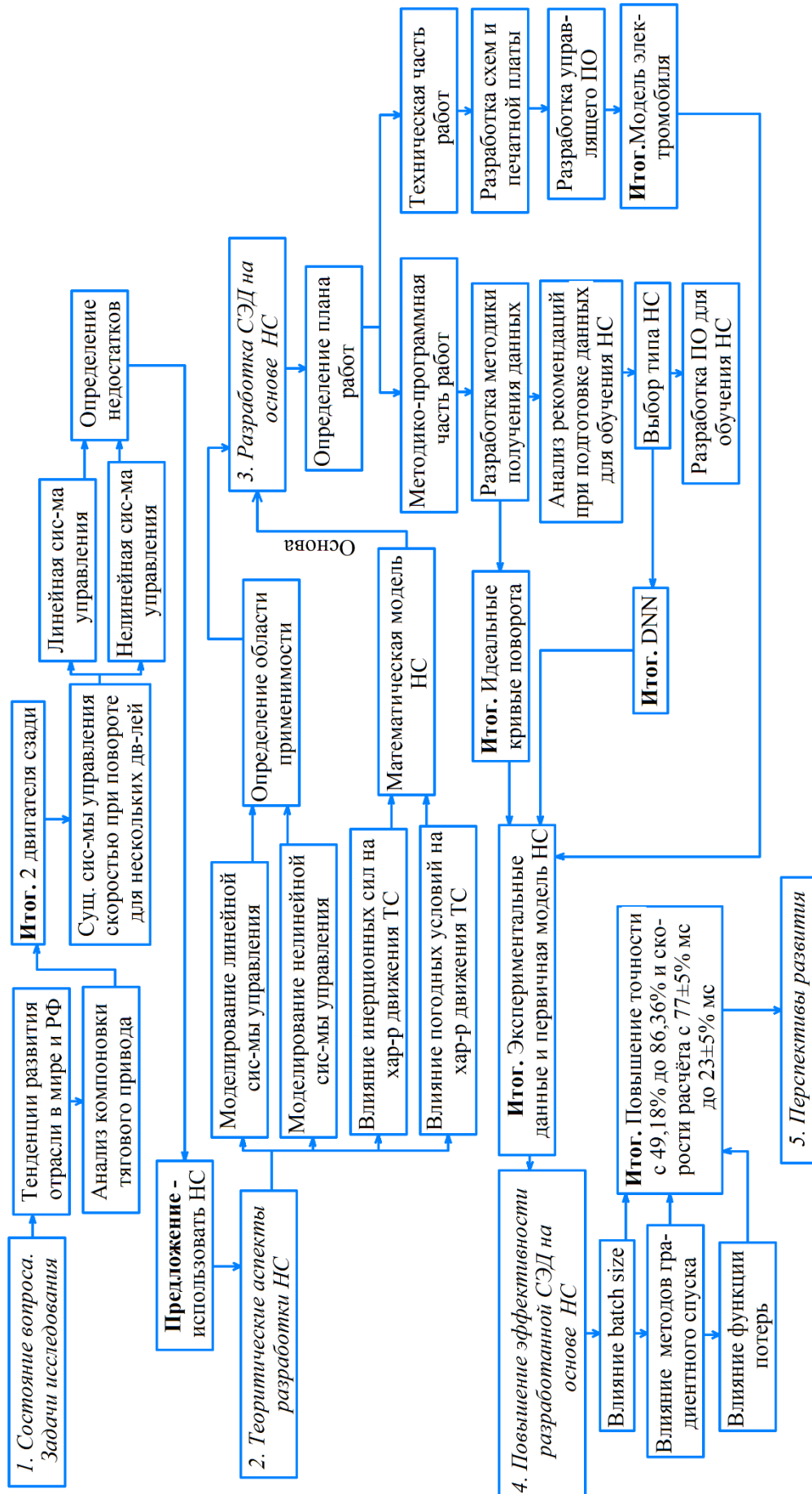
# Сделаем предсказания
train_predict = Loading_model.predict(X_train)
test_predict = Loading_model.predict(X_valid)
# Среднеквадратичное отклонение
print_error(y_train, y_valid, train_predict, test_predict)

36 ячейка. Проверка скорости работы
data = [[1, 0.9529, -0.0297, -0.0028, 0.6945, 0, 0, 0, 0.8227, 0]]
data_2 = [[0.9259, 0.7549, 0.0609, -0.0205, 0.539, 0, 0, 0, 0.7947, 0.3]]
data_3 = [[0.9259, 0.7353, 0.1244, -0.04, 0.4651, -0.0628, -0.0628, -
0.0628, 0.944, 0.075]]
start_time = time.time()
Loading_model.predict(data)
Loading_model.predict(data_2)
Loading_model.predict(data_3)
time_for_operation = (time.time() - start_time)
print(f"Среднее время на расчёт = {round(time_for_operation / 3, 5)} секунд")

37 ячейка. Скорость расчёта при анализе пакета в 22 значения тестовой выборки
start_time = time.time()
predict_arr = np.array(X_valid, ndmin=2)
test_predict = Loading_model.predict(predict_arr)
time_for_operation = (time.time() - start_time)
print(f"Время на расчёт = {round(time_for_operation / 3, 5)} секунд")

```


Приложение Д. Дорожная карта диссертационной работы



РОССИЙСКАЯ ФЕДЕРАЦИЯ



СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2024611351

Нейронная сеть для обучения электронной
дифференциальной системы на экспериментальных
данных

Правообладатель: *Лисов Андрей Анатольевич (RU)*

Автор(ы): *Лисов Андрей Анатольевич (RU)*

Заявка № 2023689565

Дата поступления 25 декабря 2023 г.

Дата государственной регистрации

в Реестре программ для ЭВМ 19 января 2024 г.



Руководитель Федеральной службы
по интеллектуальной собственности

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат 429b6a0fe3853164baf96f83b73b4aa7
Владелец **Зубов Юрий Сергеевич**
Действителен с 10.05.2023 по 02.08.2024

Ю.С. Зубов

Приложение Ж. Акт внедрения в производственный процесс

 УТВЕРЖДАЮ
главный конструктор ООО «ДСТ-Урал»
Вансович Е.И.

АКТ

Использования результатов диссертационного исследования
Лисова Андрея Анатольевича на тему: «Разработка и исследование системы
электронного дифференциала для электромобиля»

В работе рассмотрены вопросы реализации функции механического дифференциала для электромобилей с бортовым приводом ведущих мотор-колес.

Предложено и реализовано в программных кодах электронное управление тяговыми электродвигателями с помощью нейронной сети.

На тракторном заводе «ДСТ-Урал» предполагается использовать результаты диссертационной работы Лисова А.А. при проектировании колесного погрузчика с электрической трансмиссией.

Ведущий конструктор «ДСТ-Урал»  Дубровский Н.В.