МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

На правах рукописи

Фархуд Азхар Кадим Фархуд

МЕТОДЫ И АЛГОРИТМЫ ПЛАНИРОВАНИЯ ТРАЕКТОРИЙ ДВИЖЕНИЯ РОБОТОВ В ДВУМЕРНЫХ КАРТОГРАФИРОВАННЫХ СРЕДАХ

2.5.4 Роботы, мехатроника и робототехнические системы

Диссертация на соискание ученой степени
кандидата технических наук

Научный руководитель: доктор технических наук, доцент,
Медведев Михаил Юрьевич

Таганрог – 2023

СОДЕРЖАНИЕ

Введение …… 5
Глава 1. Современные подходы к планированию пути подвижных объектов 13
1.1. Описание задачи планирования пути …… 13
1.2. Классические подходы к планированию пути ……. 15
1.2.1 Клеточная декомпозиция пространства …… 15
1.2.2 Искусственные потенциальные поля и неустойчивые режимы для планирования траекторий подвижных объектов ……. 17
1.2.3 Методы планирования пути в дискретном пространстве …. 22
1.3. Применение интеллектуальных эвристик для планирования пути ……. 26
1.3.1. Планирование пути с применением генетических алгоритмов …….. 27
1.3.2. Планирование пути с применением нечеткой логики ……. 29
1.3.3. Планирование пути с применением нейронных сетей ……. 31
1.3.4. Планирование пути с применением биоинспирированных эвристик …….. 36
1.4. Выводы по главе 1 …….. 36
Глава 2. Двухэтапный метод планирования глобальной траектории движения робота ……. 38
2.1. Постановка задачи …….. 38
2.2. Алгоритм нанесения априорно неизвестных препятствий на карту по данным СТЗ ……. 43
2.3. Алгоритм планирования траектории и управления движением мобильного робота ……... 45
2.3.1 Синтез алгоритма изменения скорости движения …… 45
2.3.2 Синтез алгоритма изменения направления движения …….. 49

2.3.3 Оптимизация и сглаживание траектории движения …….. 64
2.4 Исследование алгоритмов планирования численными методами ……. 67
2.4.1 Исследование влияния числа узлов на длину и время расчета траектории …….. 68
2.4.2 Исследование влияния параметров локальной оптимизации на планируемую траекторию …… 71
2.4.3 Моделирование движения робота в среде с препятствиями 72
2.5 Выводы по главе 2 …… 77
Глава 3. Планирование движения с использованием нейронных сетей …… 78
3.1. Общее описание проблемы планирования движения с использованием нейросетевых технологий ……... 78
3.2. Постановка проблемы планирования движения на базе нейронной сети глубокого обучения …….. 79
3.3. Итерационное создание обучающей выборки и обучение нейронной сети ……. 82
3.4. Исследование итерационного алгоритма методом численного моделирования …….. 87
3.5. Применение нейронной сети для выбора алгоритма планирования ……... 93
3.6. Выводы по главе 3 …… 94
Глава 4. Экспериментальные исследования разработанных методов и алгоритмов …… 96
4.1. Описание объекта исследования ……... 96
4.2. Проведение экспериментов с использованием симулятора … 98
4.3. Проведение экспериментов с использованием колесного робота ……. 106

4.4. Выводы по главе 4 …… 109
Заключение 111
Список источников …… 113
Приложения …… 136

Введение

Проблема планирования движения мобильных роботов является актуальной на протяжении многих десятилетий. Особенную важность данная проблема приобретает в связи с необходимостью повышения степени автономности до уровня, позволяющего исключить оператора из контура управления движением. Большинство известных методов планирования движения в той или иной мере базируются на известной (возможно, частично) карте окружающей среды. Для такой постановки задачи известно большое количество методов и алгоритмов планирования траектории движения, которые обладают как известными преимуществами, так и ограничениями. Основную сложность вызывает неопределенность среды, которая заключается в ее динамическом изменении, при котором планируемая траектория должна на каждом шаге обновляться. В условиях, когда необходимо также учитывать большое количество ограничений, связанных с конструктивными и динамическими ограничениями самих роботов, процесс планирования может занимать много времени, предъявлять повышенные требования к занимаемой памяти. В этой связи актуальным вопросом развития методов планирования являются снижение времени на расчет траектории, обеспечивающей оптимизацию некоторого критерия качества.

Различным вопросам и аспектам проблемы повышения эффективность планирования движения мобильных роботов посвящены работы как отечественных (Гайдук А.Р., Веселов Г.Е., Каляев И.А., Капустян С.Г., Лохин В.М., Макаров И.М., Мартыненко Ю.Г., Медведев М.Ю., Мунасыпов Р.А., Павловский В.Е., Платонов А.К., Пшихопов В.Х., Рубцов И.В., Черноусько Ф.Л., Юревич Е.И., Ющенко А.С. и др.), так и зарубежных (Borenstein J., Brooks R.A., Dijkstra E.W., Dorigo M., Khatib O., Koditschek D.E., Korf R.E., Latombe J.-C., LaValle S.M., Stentz A., Shibata T., Zhang X. и др.) ученых.

Из существующих работ можно сделать вывод, что для компенсации неопределенности карты среды, учета ее динамики и различных ограничений наиболее перспективными являются многокомпонентные алгоритмы. Это обусловлено тем, что не существует алгоритма планирования, который одинаково эффективен во всех возможных условиях применения. Также работы известных ученых говорят о том, что целесообразно разделять алгоритмы планирования траекторий на глобальные и локальные. Глобальная траектория строится на основе априорной карты, а локальные алгоритмы используются для ее коррекции на основе данных систем технического зрения. Вместе с тем, в условиях существенного изменения среды, локальные алгоритмы планирования не дают оптимальных траекторий движения. Кроме того, локальной коррекции траектории может оказаться недостаточно, т.к. в динамической среде часто возникают ситуации, в которых возврат на исходную траекторию движения оказывается нецелесообразным. В этой связи возникает необходимость совершенствования методов глобального планирования таким образом, чтобы их быстродействия было достаточно для компенсации неопределенности и динамичности среды.

Основными механизмами компенсации указанных факторов являются интеллектуальные эвристики и поисковые вариации, основанные, например, на случайных процессах. Таким образом, повышение быстродействия методов глобального планирования и их совершенствование для динамических неопределенных сред, а также учета разнообразных ограничений, находится в фокусе современных исследований.

Объектом исследования является мобильный робот, функционирующий в двумерной картографированной среде.

Предметом исследования являются методы и алгоритмы планирования траекторий движения мобильных роботов в двумерной картографированной среде.

Целью настоящей работы является развитие методов и алгоритмов глобального планирования траекторий движения в двумерных картографированных средах, обеспечивающих снижение времени расчета, длины планируемого пути и возможность применения в динамических средах.

Научная задача, решение которой содержится в диссертационной работе – разработка, обоснование и исследование методов и алгоритмов глобального планирования движения в картографированных средах, обеспечивающих повышение быстродействия при планировании, а также учет динамических изменений среды и ограничений, накладываемых свойствами мобильных роботов.

Основные задачи диссертационного исследования:

– обзор и анализ известных методов планирования движения мобильных роботов;

– разработка и исследование двухэтапного метода планирования движения, обеспечивающего быстрое построение траектории начального приближения и ее оптимизацию локальными методами;

– исследование известных алгоритмов глобального планирования и их модификация с целью повышения времени расчета получаемой траектории;

– разработка метода учета ограничений, обусловленных конструктивными и динамическими свойствами мобильных роботов;

– разработка метода интеллектуального планирования траекторий движения мобильных роботов, обеспечивающего функционирование в динамических и неопределенных средах;

– разработка метода обучения интеллектуального планировщика движения, обеспечивающего снижение объема результирующей обучающей выборки при сохранении качества обучения;

– проведение экспериментальных исследований, подтверждающих достижение поставленной в диссертации цели.

Методологическую основу диссертационной работы составляют методы планирования движения на основе вероятностных подходов, методы клеточной декомпозиции, методы поиска оптимальных путем на графах, многоэтапные методы решения задач с получением грубого и точного решений, методы контролируемого обучения, теория обыкновенных дифференциальных уравнений, матричное исчисление, численное моделирование и натурные эксперименты.

Достоверность результатов диссертации:

– обеспечивается использованием математических положений и выводов, основанных на матричном анализе и теории оптимизации;

– подкрепляется полученными результатами натурных и численных экспериментальных исследований;

– не противоречат и согласуются с известными результатами исследований по данной тематике российских и зарубежных ученых .

Полученные автором диссертации научные результаты, обладающие научной новизной:

1. Двухэтапный метод планирования траектории движения робота, **отличающийся** использованием на первом этапе модифицированного алгоритма быстро растущих случайных деревьев для получения траектории первого приближения, а на втором этапе – итерационного алгоритма локальной оптимизации. Применение данной процедуры позволяет снизить вычислительные затраты за счет того, что алгоритм быстро растущих случайных деревьев применятся только для получения грубой траектории первого приближения, что может быть сделано достаточно быстро. Процедура сглаживания и оптимизации траектории осуществляется более быстрыми известными алгоритмами локальной оптимизации.

2. Модифицированный алгоритм быстро растущих случайных деревьев, **отличающийся** использованием нескольких родительских узлов, что позволяет сократить время расчета траектории на 20 – 25 % по сравнению с

оригинальным алгоритмом и при одинаковой степени приближения к оптимальной траектории.

3. Метод учета ограничений на маневры робота, зависящие от его скорости, **отличающийся** использованием виртуальных препятствий, которые перекрывают физически недоступные для движения сектора, что позволяет учитывать динамические свойства мобильного робота без усложнения алгоритмов планирования, ориентированных на использование карты среды.

4. Метод планирования траектории подвижного объекта, **отличающийся** применением сети глубокого обучения, которая вычисляет направление движения в текущий момент времени. В качестве учителя используется алгоритм D*, в результате чего последовательность выбранных обученной нейронной сетью направлений оказывается близка к траектории, формируемой D*. Предложенный метод обеспечивает уменьшение времени планирования по сравнению с D*, в среднем на 30 % для двумерной среды с плотностью заполнения препятствиями до 50 %. Указанное сокращение времени расчета получено для неопределенных и динамических сред.

5. Метод поэтапного обучения глубокой нейронной сети, **отличающийся** итерационной процедурой, в ходе которой на каждой итерации в обучающую выборку добавляются примеры, в которых сеть принимает решения, отличающиеся от учителя. Предложенный метод обеспечивает снижение размера обучающей выборки до 40 % по сравнение с обычным созданием обучающих примеров. Данная процедура создания обучающей выборки хорошо автоматизируется, что позволяет уйти от ручной обработки.

Основные положения, выносимые на защиту:

1. Планирование грубой траектории первого приближения методами глобального поиска на первом этапе и оптимизация полученной траектории первого приближения на втором этапе обеспечивает более высокое быстродействие по сравнению с планированием траектории методами глобального поиска.

2. Использование нескольких родительских узлов в алгоритме быстрорастущих случайных деревьев позволяет сократить время планирования при одинаковой степени оптимальности траектории движения.

3. Введение виртуальных препятствий на карту среды позволяет учесть ограничения на маневры, связанные со скоростью движения робота, без изменения алгоритма планирования.

4. Нейронная сеть параллельно обрабатывает последовательный расчет траектории движения, что позволяет не рассчитывать в явном виде траекторию, а вырабатывать направление движения в текущий момент времени.

5. Итерационная процедура, в ходе которой в обучающую выборку добавляются примеры, в которых обученная сеть принимает решения, отличающиеся от учителя, обеспечивает снижение размера обучающей выборки без снижения качества обучения.

Теоретическая и практическая значимость диссертационной работы. Теоретическая значимость полученных результатов заключается в развитии методов и алгоритмов планирования движения в частично картографированных динамических средах.

Практическая значимость полученных результатов определяется расширением функциональных возможность методов планирования, которые заключаются в учете ограничений на маневры мобильных роботов, а также снижением времени, затрачиваемого на расчет траектории движения.

Все разработанные алгоритмы реализованы на скриптовом языке MATLAB в виде программ, которые могут компилироваться для дальнейшего применения в бортовых системах планирования движения мобильных роботов.

Предложенные алгоритмы характеризуются вычислительной сложностью, которая может быть реализована на современных бортовых микроконтроллерах.

Апробация работы. Основные положения диссертации были доложены на 22 Всероссийской научно-технической конференции с международным участием имени профессора О.Н. Пьявченко «Компьютерные и информационные технологии в науке, инженерии и управлении» 2020 г., II Всероссийской молодежной научно-практической конференции студентов, аспирантов и молодых ученых «Наука и творчество: вклад молодежи» 2021 г., 7-й Международной конференции по мехатронике и робототехнике (7th International Conference on Mechatronics and Robotics Engineering) 2021 г., Всероссийской научно-технической конференции с международным участием имени профессора О.Н. Пьявченко «Компьютерные и информационные технологии в науке, инженерии и управлении» 2022 г., III Международной научно-практической конференции «Инновационные направления интеграции науки, образования и производства» 2022 г., II Международной научно-практической конференции «Научная инициатива иностранных студентов и аспирантов» 2022 г., Международной научно-практической конференции «Автомобиле- и тракторостроение»2022 г.

Реализация и внедрение результатов работы. Двухэтапный метод планирования траектории движения подвижного объекта, модифицированный алгоритм быстро растущих случайных деревьев и метод учета ограничений с использованием виртуальных препятствий, полученные в диссертации, использованы при выполнении гранта РНФ 22-29-00533, выполняемого в ОАО «НКБ робототехники и систем управления».

Метод планирования движения на основе нейронной сети глубокого обучения и поэтапный метод обучения внедрены в учебный процесс кафедры электротехники и мехатроники Института радиотехнических систем и управления Южного федерального университета.

Личный вклад автора. Результаты, обладающие научной новизной и основные положения, выдвигаемые для защиты, получены автором диссертационного исследования лично.

Публикации. Основные результаты опубликованы в 3-х статьях в ведущих научных изданиях, рекомендованных ВАК для публикации результатов работ по диссертациям на соискание ученой степени кандидата технических наук, 1-й статье в иностранных научных изданиях, включенных в систему цитирования Scopus, 6-и докладах в материалах всероссийских и международных конференций.

Структура и объем работы. Диссертационная работа состоит из введения, четырех глав, включающих обзор и постановку задачи, разработку методов и алгоритмов планироования, результаты численных и экспериментальных исследований, библиографический список из 203 наименований, содержание и два приложения. Основная часть работы изложена на 135 страницах и включает в себя 49 рисунков и 9 таблиц.

Глава 1. Современные подходы к планированию пути подвижных объектов

1.1. Описание задачи планирования пути

Термин «планирование пути» используется во многих областях, таких как робототехника, искусственный интеллект или теория управления. В различных областях применяются различные определения этого термина [1, 2]. В робототехнике планирование пути связано с проблемой перемещения робота из одной точки в другую. С развитием робототехники задача планирования пути стала включать в себя такие аспекты, как учет неопределенности [3, 4], групповое планирование пути [5 – 9], учет динамики среды [10, 11]. В области искусственного интеллекта под планированием пути понимается поиск последовательности логических действий, которые преобразуют начальную конфигурацию робота в желаемую целевую конфигурацию. Такое планирование может базироваться на процессах принятия решений Маркова [12, 13], информации о несовершенном состоянии [1], методах обучения [14, 16, 17] или теоретико-игровых моделях [15, 18]. Управление движением включает вопросы планирования траекторий, оптимальных по заданному критерию [4, 22 – 24], обеспечения устойчивости запланированной траектории [16, 19], а также формирования управления вдоль запланированной траектории [20, 21]. Указанные аспекты определяют широту проблемы планирования траекторий движения, которая решается большим числом методов. Далее, рассмотрим основные из этих подходов и методов.

Значительное число методов планирования траекторий подвижных базируется на алгоритмах, учитывающих геометрическую модель объекта и среды [1]. При этом задача планирования траектории движения может

разделена на планирование движения объекта, планирование его траектории и компенсацию неопределенности среды. Согласно [1] проблемы планирования оптимальных траекторий и обеспечения устойчивого движения относительно них являются центральными при разработке систем управления подвижными объектами. Как правило, методы планирования учитывают кинематику подвижного объекта. Динамика, при этом, может учитываться в простейшем виде. Однако повышение эффективности систем управления движением связано с учетом неопределенности среды [25], динамических ограничений [10, 11, 26] и достижением глобального оптимума планируемых траекторий [22, 23, 27]. Планирование траектории включает расчет оптимальной по заданному критерию траектории в заданной или неопределенной среде и обеспечение устойчивого движения подвижного объекта относительно запланированной траектории. При этом возможен учет ряда кинематических или технических ограничений [28 – 30].

При отсутствии внешних навигационных системах, наряду с задачей планирования движения, робот должен решать навигационную задачу, которая может включать построение карты окружающей среды [31, 32], определение его точного места на карте [32, 33] или прибытие в определенное место без столкновения [34 – 36]. Первые две задачи обычно объединяются в одну задачу, называемую одновременной локализацией и отображением (SLAM) [33, 37 – 42]. Эта проблема является хорошо известной, и в настоящее время существует множество решений с открытым исходным кодом, например, HectorSLAM [40], OpenSLAM [41] и т.д.

Навигация робота в окружающей среде зависит от наличия карты. Если у робота нет карты среды, он использует свои датчики. На основе показаний датчиков он может перемещаться из одного места в другое. Такой подход называется реактивной навигацией, потому что робот реагирует только на сигналы датчиков [43]. Однако, если у робота есть точная карта

окружающей среды, он может использовать множество методов для оптимальной навигации. Такой процесс называется планированием пути или глобальной навигацией. Правильный выбор метода навигации является наиболее важным шагом в планировании траектории робота при работе в простой и сложной среде. В настоящее время различными исследователями в области навигации мобильных роботов разработано множество методов.

В течение нескольких десятилетий различные исследователи и ученые предлагали многочисленные методологии планирования пути. Методы, используемые для планирования движения мобильных роботов, в целом, подразделяются на две категории: классические и эвристические подходы.

1.2. Классические подходы к планированию пути

Основными классическими подходами в настоящее время являются подходы, базирующиеся на клеточной декомпозиции и последующем применении графоаналитических методов поиска пути, вероятностных подходов или метода потенциальных полей.

1.2.1 Клеточная декомпозиция пространства

Как правило, планирование пути осуществляется в конфигурационном пространстве [44 – 46], которое включает в себя множество препятствий, кинематические и динамические ограничения и механическую систему робота, как совокупность точек или твердых тел. В этой связи конфигурация понимается как множество переменных состояния, которое однозначно определяет положение робота в пространстве. Клеточная декомпозиция [45] является базой для значительного числа методов планирования пути и представляет собой разбиение исходного непрерывного пространства на дискретные ячейки, например, согласно процедуре [47]. После декомпозиции поиск пути осуществляется уже в дискретном пространстве, которое описывает исходное пространство с определенной точностью. В

общем случае, целом, разделение пространства на ячейки в оптимальном по точности смысле требует различного размера ячеек в различных областях этого пространства. В этой связи широко применяются методы адаптивной декомпозиции [48]. Основной проблемой методов, базирующихся на клеточной декомпозиции, является увеличение вычислительной сложности и требуемого объема памяти с ростом требований к точности планирования пути.

Как правило, поиск пути в дискретном пространстве представляет собой граф связности для последовательного перехода от одной ячейки к другой до достижения целевой точки. Полагается, что путь от начальной до конечной ячейки существует. При точной декомпозиции [51] форма и размеры клетки определяются по карте окружающей среды, форме и расположению препятствий внутри нее. Этот метод не использует регулярную сетку. Каждая клетка при планировании считается узлом графа. При этом узлы могут быть соединены друг с другом ребрами. При точной декомпозиции границы объекта должны точно совпадать с границами ячеек. При приближенной декомпозиции [52] выполнение данного условия не требуется, поэтому возможно использование регулярной сетки и ее легче реализовать. При адаптивном разбиении пространства оно делится на крупные ячейки. Если на пути встречается занятая ячейка, то она разбивается на более мелкие части [49, 50]. Такой подход плохо подходит для планирования пути в динамической среде, т.к. при изменении положения препятствия необходимо осуществлять разбиение пространства на ячейки заново [53].

Также существует достаточно большое число эвристик, используемых для разбиения пространства на ячейки с целью дальнейшего планирования пути. Так, в работе [54] для разбиения пространства на ячейки применена гармоническая функция. При этом рассмотрены достаточно сложные препятствия S-образной формы. В статье [55] стратегия декомпозиции ячеек позволяет учитывать информацию, получаемую от сенсорных систем

мобильного робота. Для статических сред данный подход показал достаточно хорошие результаты. Аналогичный подход, базирующийся на показаниях датчиков, предложен в [56] для декомпозиции рабочей области для задачи мобильного робота на ячейки. При этом предполагается, что робот оснащен системой лазерного сканирования. В [57] предложен адаптивный алгоритм клеточной декомпозиции, основанный на искусственных потенциальных полях, что позволило снизить вычислительные затраты.

Таким образом, задача декомпозиции пространства на дискретные ячейки остается актуальной в части адаптации к неопределенности среды и повышения вычислительной эффективности методов планирования.

1.2.2 Искусственные потенциальные поля и неустойчивые режимы для планирования траекторий подвижных объектов

Метод потенциалов для планирования траекторий подвижных объектов впервые был предложен А.К. Платоновым в трудах Института прикладной математики АН СССР. В работах [58, 59] представлен метод потенциалов в задаче выбора пути. Данный метод приобрел мировую популярность после выхода работ Хатиба [60, 61].

Как известно, в методе потенциальных полей целевая точка, в которую должен двигаться робот, обладает положительным потенциалом. Препятствия и подвижный объект обладают отрицательным потенциалом. Положение целевой точки и препятствий известны. При такой постановке задачи подвижный объект, обладающий отрицательным зарядом, притягивается к целевому положению и отталкивается от препятствий. Если среда не содержит локальных минимумов, то подвижный робот в такой задаче успешно достигает целевой точки. Подробный обзор методов планирования пути, основанных на потенциальных полях, можно найти в [4]. Здесь отметим основные достоинства и недостатки метода. Применение метода потенциалов в составе многокомпонентных подходов будет рассмотрено позже.

Метод потенциальных полей позволяет с высокой вычислительной эффективностью двигаться в среде с препятствиями. Однако, при наличии неопределенностей вид траектории трудно прогнозируем и сильно зависит от выбранных функций отталкивания и притягивания. Также, метод потенциальных полей может приводить робот в так называемые локальные минимумы. Выбор наиболее подходящих функций притягивания и отталкивания, учитывающих различные ограничения и характеристики робота, а также решение проблемы локальных минимумов являются центральными вопросами, которым посвящены различные исследования.

В первых работах О. Хатиба предполагалось отсутствие ошибок навигационной системы, позволяющей точно измерять положение робота в глобальной системе. Также робот описывался кинематикой движения материальной точки.

В ходе развития метода исследования проводились по нескольким направлениям. Первое направление связно с выбором силовых функций таким образом, чтобы отсутствовали точки равновесия. В работах [62, 63] различные функции отталкивания, позволяющие эффективно обходить препятствия и избегать локальных минимумов. В [62] для обхода препятствий в неопределенной среде предложена гармоническая функция отталкивания. Показано, что ее использование позволяет избежать локального минимума. В [63] также исследуется проблема попадания робота в локальные минимумы. При этом рассмотрены динамические свойства робота. Этому направлению также посвящены работы [64, 65], в которых суммарная отталкивающая сила вычисляется как произведение отталкивающих сил каждого препятствия, что, при определенных условиях, позволяет избежать появления точек равновесия. Однако авторам работ [64, 65] не удалось обобщить постановку задачи и избежать получения негладких траекторий. В работе [66] функция отталкивания формируется с учетом расстояния до цели, что позволяет расположить глобальный минимум потенциального поля в

целевой точке. В статье [67] предложено формировать случайные вариации потенциального поля при детектировании попадания подвижного объекта в локальный минимум. В [68] на основе проведенного анализа предлагается на основе анализа текущей ситуации изменять отталкивающие и притягивающие потенциальные функции. В [69] также рассмотрена задача избегания локального минимума. Для этого предложена концепция превосходящего отталкивающего потенциала, что позволило при заданных ограничениях выводить робот из локальных минимумов.

Еще одно направление исследований потенциальных полей посвящено учету различных свойств мобильных роботов, а также учету геометрии, ориентации и динамики. Это позволяет учитывать при планировании траектории движения особенности конструкции и управляемость робота, ограничения на скорости и ускорения. В работе [70] робот и препятствия среды представлены не материальными точками, а выпуклыми многоугольниками. Для этих условий предложен алгоритм планирования движения. Выход из локальных минимумов осуществляется за счет эвристического алгоритма, базирующего на изменении угла поворота подвижного объекта. В статье [71] потенциальное поле определяется с учетом кривизны траектории движения, что позволяет учесть динамические ограничения, накладываемые на подвижный объект. В работе [72] разработан механизм управления скоростью в методе потенциальных полей, что позволяет учесть ряд динамических свойств робота и препятствия при достижении цели.

Третье направление исследований связано с адаптацией метода искусственного потенциального поля для локальной коррекции пути при обнаружении роботом препятствий [73 – 76]. В работе [73] представлен метод обхода статичных препятствий, которые описываются множеством точек на плоскости. При этом используется модифицированный алгоритм потенциальных полей. В [74] исследуются различные локальные алгоритмы плани-

рования, которые включают искусственные потенциальные поля. Исследование ориентировано на решение задачи планирования пути мобильного робота в реальном времени. В предложенном авторами методе осуществляется предварительное картографирование локальной области робота. В работе [75] исследован алгоритм коррекции траектории робота в области препятствий, основанный на использовании гистограмм векторных полей, который за счет локального картографирования позволяет оперативно обходить препятствия во время движения. В [76] решается проблема устранения колебаний при навигации мобильных роботов с помощью искусственного потенциального поля. Представлена улучшенная версию искусственных потенциальных полей, которая позволяет свести к минимуму колебания в ситуации, когда цель находится близко к препятствию.

Четвертое направление исследований связано с разработкой многокомпонентных методов планирования траекторий движения с привлечением интеллектуальных эвристик и искусственных потенциальных полей. В частности, в работе [70] вводятся вспомогательные точки, а направление движения выбирается с помощью генетических алгоритмов. В [77] при планировании используется метод потенциальных полей и оценка и прогнозирование ситуации, построенные на базе мультиагентных технологий. Данный метод исследуется на примере симулятора гонок автомобилей. В [78] предлагается использовать вместе с методом потенциальных полей набор эвристических правил, позволяющий обнаруживать и обходить локальные минимумы, возникающие на пути робота. В статье рассмотрен случай использования узконаправленного локатора и стационарных и нестационарных препятствий.

В работах [100 – 103] предлагается формировать динамические отталкивающие и притягивающие силы, т.е. вместо потенциальных сил, зависящих от координат подвижного объекта и препятствия $F(y)$, где y – вектор координат; использовать зависимости вида:

$$F_{ar}\left(y,\dot{y},\ddot{y},\beta\right)=0, \tag{1.1}$$

где $F_{ar}\left(y,\dot{y},\ddot{y},\beta\right)$ – функция, в зависимости от значения параметра β, являющаяся притягивающей или отталкивающей.

В работах [100, 101] применяется третья теоремя Ляпунова (теорема о неустойчивости) для линейного дифференциального, задающего динамику замкнутой системы управления относительно желаемой траектории движения ψ(y):

$$\ddot{\psi}+T_1\left(\beta\right)\dot{\psi}+T_2\left(\beta\right)\psi=0, \tag{1.2}$$

где $T_1\left(\beta\right), T_2\left(\beta\right)$ – матрицы коэффициентов.

Водится бифуркационный параметр вида:

$$\beta=\sum_j\left|r_j-r_s\right|-\sum_j\left(r_j-r_s\right),$$

где r_s – безопасное расстояние до препятствия; r_j – текущее расстояние до j-го препятствия.

Из последнего выражения следует, что если расстояния от подвижного объекта до препятствий больше допустимого значения r_s, то бифуркационный параметр β равен нулю, а матрицы $T_1\left(\beta\right), T_2\left(\beta\right)$ положительно определенные. В этом случае положение равновесия ψ(*y*)=0 является устойчивым. Если одно из расстояний r_j становится меньше r_s, то бифуркационный параметр β больше нуля, а матрицы $T_1\left(\beta\right), T_2\left(\beta\right)$ становятся отрицательно определенными. В этом случае положение равновесия ψ(*y*)=0 является неустойчивым.

Как и метод потенциальных полей, метод неустойчивых режимов имеет ограничения на применение в сложных средах, связанные с наличием локальных минимумов. В этой связи в [102, 103] предложен гибридный алгоритм, сочетающий неустойчивые режимы с методом виртуальной целевой точки, что позволило авторам решить проблему попадания в локальные

минимумы в условиях, когда статические препятствия образуют сложные конфигурации. Пропорционально бифуркационному параметру (3) изменяется направление движения на угол $\gamma = k_\gamma \beta$.

В работах [102, 103] предложены различные алгоритмы децентрализованного управления группой подвижных объектов, в которых для обхода препятствий используются динамические силы, формируемые дополнительными дифференциальными уравнениями. Это позволяет функционировать группе роботов в неопределенной среде с препятствиями. Для обхода препятствий при движении группы подвижных объектов в трехмерной среде предлагаются различные варианты новых гибридных алгоритмов, включающих в себя отталкивающие динамические силы, генерируемые неустойчивой системой уравнений и интеллектуальный алгоритм определения направления обхода препятствия. Дополнительно предложен метод формирования нерегулярного строя на основе триангуляции Делоне.

1.2.3 Методы планирования пути в дискретном пространстве

Из данного класса алгоритмов широко известны алгоритмы A* и D* [79, 80]. Базовым алгоритмом является алгоритм Дейкстры, который обладает большой вычислительной сложностью. Для ее снижения в алгоритме A* используется специальная эвристическая функция оценки стоимости пути. Как и алгоритм Дейкстры, A* использует большой объем памяти, а его вычислительная сложность быстро растет с увеличением количества ячеек на карте. В связи с этим существуют модификации A*, направленные на устранение этих недостатков. Такие модификации включают A* с итеративным углублением [81], A* с ограничениями памяти [82], иерархический A* [83], A* с динамическим изменением веса ребер [84, 85]. Алгоритм D* [80] обеспечивает планирование пути в условиях, когда информация об окружающей среде может обновляться во время движения подвижного объ-

екта. Он может эффективно применяться с обновляющейся картой. Алгоритмы A*, D* дают траекторию в виде последовательности прямых линий. В связи с этим эти методы дополняются дополнительными алгоритмами, которые используются для сглаживания пути [86]. Учет динамических ограничений, налагаемых инерцией робота, также является отдельной трудоемкой задачей.

В настоящее время широко используются вероятностные методы планирования траекторий в дискретном пространстве [87, 88]. Методы этого типа включают метод вероятностной дорожной карты (PRM) [44, 89 – 93] и метод быстрого изучения случайных деревьев (RRT) [1, 94 – 97]. Метод PRM обычно не учитывает динамику мобильного робота. Для его рассмотрения требуются значительные вычислительные затраты [90, 98]. Метод RRT может учитывать динамику мобильного робота [94, 95], поскольку он строит физически реализуемый путь в свободном пространстве. Основными ограничениями этих методов являются медленная сходимость к оптимальному решению, высокие требования к используемой памяти [90, 98] и ограничения в планировании в условиях неполного определения модели робота и окружающей среды [98].

Проблема использования планировщиков RRT и RPM в условиях частичной неопределенности рассмотрена в [90], в которой предлагается разделить всю область эксплуатации на отдельные регионы. Поиск связей между регионами выполняется с помощью метода RRT или RPM. Планирование локального маршрута в пределах определенного региона осуществляется с использованием традиционных методов планирования. В этом случае неопределенность модели робота учитывается путем введения аддитивного случайного возмущающего входного сигнала. В качестве фактора неопределенности вводится вероятностью того, что препятствие находится в заданной ячейке среды. Рассматриваемый в [90] метод обеспечивает нахож-

дение пути, который гарантирует успешное достижение цели с вероятностью, не меньшей заданное значение pmin. В работе [90] проведено аналитическое исследование разработанного алгоритма. Также представлены результаты численного моделирования системы управления подвижного объекта, описываемого кинематикой материальной точки. Среда является стационарной. Основным недостатком метода является значительное увеличение вычислительных затрат при больших значениях вероятности pmin.

Разновидностями метода дорожных карт являются графы видимости [111, 112] и диаграммы Вороного [113]. Метод графа видимости используется для среды с многоугольными препятствиями, в которой вершины многоугольника представлены узлами, которые соединены межу собой ребрами. Метод диаграмм Вороного делит область на субрегионы, центры которых равноудалены от соседних двух точек на границах препятствия. Основные ограничения метода дорожных карт – высокие вычислительные затраты и большие объемы требуемой памяти.

Проблема снижения вычислительных затрат и занимаемой памяти рассмотрена в [96, 98], в которой объединены методы RRT и потенциальных полей (P - RRT*). В рассматриваемом алгоритме направление построения случайного дерева определяется градиентом потенциального поля. Это для некоторых сред позволяет увеличить сходимость вычисляемой траектории к оптимальному решению. В статье [98] предложена модифицированная версия алгоритма RRT*, в которой начальный путь вычисляется аналогично базовому алгоритму RRT. Затем количество узлов в пространстве конфигурации увеличивается для оптимизации начального пути. В [99] проведен анализ сходимости предложенного алгоритма и представлены результаты численного моделирования. В [96] метод PRRT* разработан для случая двунаправленного поиска. В данном алгоритме строится два дерева,

одно из начальной точки, а второе целевой точки. При этом оба дерева создают притягивающие поля, что улучшает сходимость метода PERT*, особенно для поиска узких проходов.

Обратим внимание, что идея использования потенциальных полей для повышения сходимости и вычислительной эффективности не ограничивается только семейством алгоритмов RRT, но также используется с алгоритмами PRM. Так, в [91] используется гибридный метод, включающий алгоритм PRM и искусственные потенциальные поля. Распределение узлов использует сегментированную карту для создания областей с низким и высоким потенциалом. Для устранения проблемы локальных минимумов используется реактивный подход к планированию локального пути при наличии статических и динамических препятствий. Предложенные алгоритмы были протестированы с использованием численных и полевых экспериментов. В работе [104] также исследуется проблема повышения скорости сходимости процесса планирования маршрута с использованием метода RRT. В этой работе предложена двухэтапная процедура, в ходе которой метод RRT используется для поиска пути первого приближения, который оптимизируется с использованием разработанной улучшенной версии алгоритма RRT*. В указанной статье также предлагается подход, который повышает эффективность планирования маршрута в динамичной среде. Согласно этому подходу, если необходимо перепланировать путь, рассматривается не все дерево целиком, а только его часть, которая описывает оптимальный путь. Эффективность предложенных алгоритмов продемонстрирована методами численного моделирования.

В работе [105] также используется двухэтапная процедура планирования для динамической среды. Глобальный планировщик, используя метод RRT, планирует исходный путь первого приближения. Далее используется динамический планировщик, который изменяет узлы, базируясь на

значениях отталкивающих сил и сил инерции. Для учета движущихся препятствий в этом методе используется процедура прогнозирования. Эффективность метода подтверждена результатами численных экспериментов.

В [106] рассматривается модификация алгоритма RRT с учетом ограничений, налагаемых системой зрения, и ограничений скорости. Предложен алгоритм, основанный на компьютерном зрении, который функционирует в условиях неизвестной глубины изображения. Рассмотрена модель робота в виде кинематики материальной точки. Основной вклад метода, предложенного в [106], заключается в наличии виртуальной цели, которая расположена так, чтобы робот находился на линии, обеспечивающей заданный угол его ориентации в целевой точке. Преимущества предложенного алгоритма подтверждены результатами проведенных экспериментов. В работе [107] рассматривается задача построения оптимального пути, который должен пройти через заданное множество точек [108]. Планирование пути осуществляется путем построения не одного дерева, а леса деревьев [109]. Каждое дерево начинается в одной из целевых точек. Построение множества деревьев осуществляется до их соединения в одно дерево. Эта работа является развитием статьи [110], в которой построение леса деревьев оптимизировано за счет приоритетов очереди узлов. Предлагаемый метод исследуется в двумерной и трехмерной среде с использованием методов численного моделирования.

1.3. Применение интеллектуальных эвристик для планирования пути

В последнее время интеллектуальные и биоинспирированные эвристические подходы, такие как генетические алгоритмы, нечеткая логика, нейронные сети, методы оптимизации роя частиц, муравьиные алгоритмы, алгоритмы пчелиных колоний и многие другие, приобрели большую популярность как инструмент для решения задач навигации мобильных роботов.

Данный класса алгоритмов обладает способностью справляться с неопределенностью, присутствующей в окружающей среде. Различные эвристические подходы обсуждаются ниже.

1.3.1. Планирование пути с применением генетических алгоритмов

Генетические алгоритмы представляют собой популярный инструмент поисковой оптимизации, который следует принципу генетики и естественного отбора, впервые открытому Бремерманом [114] в 1958 году. В настоящее время он широкое применяется во всех областях науки и техники, включая планирование пути мобильных роботов. Преимуществом применения генетических алгоритмов является возможность эффективного поиска пути, обеспечивающего глобальный оптимум заданной целевой функции. Для этого используется популяция, из которой итерационно отбираются наиболее успешные особи. Для улучшения процедуры поиска глобального оптимума в популяции применяются генетические процедуры скрещивания и мутаций. Генетические алгоритмы показывают высокую эффективность в статических средах и при высокой размерности поисковых задач. В этой связи существует обширная библиография по применению генетических алгоритмов в задачах поиска пути.

Так, в работе [115] представлено исследование, посвященное применению генетических алгоритмов для решения задачи навигации мобильного робота в статической среде. Представлены результатами моделирования при наличии полигонального препятствия, которые подтверждают эффективность примененного алгоритма. Различным аспектам применения генетических алгоритмов в статических средах также посвящены работы [116 – 118].

Большое количество исследований посвящено исследованию применения генетических алгоритмов для планирования пути в средах с подвижными препятствиями. Так в работе [120] представлен генетический алгоритм, оптимизированный для работы в неопределенной среде с подвижными препятствиями. Для динамической среды наиболее часто используются составные алгоритмы, одна из частей который реализована в виде генетического алгоритма [121 – 123].

Наиболее часто генетические алгоритмы используются в задачах группового планирования траекторий движения. Для статических и динамических сред такая задача рассмотрена в работах [119, 124 – 126], в которых планируются траектории, обеспечивающие отсутствие коллизий в группе.

Базовые генетические алгоритмы обладают низкой скоростью сходимости для решения задач в реальном времени, а также не гарантируют получение оптимального решения. Кроме того, процесс определения параметров частоты мутаций и численности популяции является трудоемким и эвристическим. В этой связи ряд исследований направлен на улучшение генетических алгоритмов для решения различных задач [127, 128]. Так в [127] используется модифицированный генетический алгоритм для решения групповой задачи навигации, обеспечивающей обход внезапно появляющихся препятствий в неопределенной среде. Аналогичная задача решается в работе [128], которая посвящена исследованию влияния длины хромосомы на эффективность решения задач планирования пути.

Генетические алгоритмы находят свое применение в системах управления подводными роботами [129, 130], роботами воздушного базирования [131, 132], шагающими роботами [133].

В наибольшей степени преимущества генетических алгоритмов проявляются в статических картографированных средах. Как показано в работе

[4] в динамических средах такие алгоритмы могут применяться в многокомпонентных системах управления для нахождения пути первого приближения, который далее оптимизируется локальными методами поиска экстремумов.

1.3.2. Планирование пути с применением нечеткой логики

Концепция нечеткой логики была впервые предложена Заде [134] в 1965 году и впоследствии использовалась во различных областях исследований и разработок.

Как известно [135], нечеткие системы включают базу правил «ЕСЛИ ТО», которая строится экспертами или на основе ключевых знаний в рассматриваемой предметной области. Простота применения таких баз, а также их универсальность и возможность решения задач планирования пути без точных вычислений сделала нечеткие системы распространенным инструментом. При решении задач планирования пути производится разделение стратегии выбора направления движения на отдельные ситуации, а затем осуществляется комплексирование этих ситуаций в более сложное поведение.

Общая последовательность формирования стратегии планирования пути с использованием нечеткой логики включает следующие этапы:

– создание нечетких множеств, описывающих положение робота и препятствий к окружающей среде;

– выработка нечетких правил для простых ситуаций, например, движение к цели, обход препятствия, выход их тупика и пр.;

– выбор комбинации нечетких правил на основе текущего состояния робота и среды.

Такой подход приводит структуре системы планирования движения, которая включает отдельные модули, которые можно наращивать в зависи-

мости от решаемых задач. Выходы указанных модулей могут комбинироваться в соответствии с правилами дефаззификации. При этом входами указанных модулей могут быть как выходы сенсорной системы робота, так и выходы других модулей нечеткой системы планирования движения.

Значительную популярность методы нечеткой логики в системах управления приобрели после опубликования работ [136, 137], представивших архитектуру категоризации, в которой введены уровни приоритетов отдельных поведений, что позволяет ранжировать различные стратегии поведения при движении роботов. Система приоритетов позволяет активировать только одно поведение из базы нечетких правил. Иногда, такой подход может не приводить к рациональному движению робота. В частности, при обнаружении препятствия, сочетание поведений «Движение к цели» и «Обход препятствия», как правило дает более оптимальную траекторию, чем только одно поведение «Обход препятствия». Иными словами, полностью прекращать движение к цели при появлении препятствия, не всегда является рациональной стратегией.

В этой связи также получили распространение методы, связанные с формированием выхода в виде взвешенной комбинации отдельных нечетких правил [138 – 140].

В работе [141] представлена мультиагентная архитектура нечеткой системы управления мобильного робота, в которой нечеткое правило в текущей ситуации активируется в результате процедуры голосования. При этом выполняются те действия, которые выигрывают голосование. Такие системы, как и системы с взвешиванием активируемых правил, могут приводить к неверным действиям. Например, два активированных с одинаковыми весовыми коэффициентами правила «Движение вперед» и «Движение назад» приводят к остановке робота. В таких случаях используются ситуативные подходы к выбору правил поведения [142, 143].

Так, в работе [142] исследован ситуативный алгоритм активации нечетких правил, в котором правильность выбранного решения оценивается по текущему состоянию. В работе [143] предложен аналогичный подход, в котором используется комбинация приоритетов и степени применимости каждого правила. Как правило, такие подходы базируются на выделении основного правила поведения, нацеленного на достижение целевой точки, и отдельных вспомогательных правил, нацеленных на обход препятствий, устранение коллизий и т.д.

Известно достаточно большое число прикладных работ, в которых нечеткая логика успешно применена для обхода препятствий [144], планирования движения робота в неструктурированной статической и динамической среде, с устранением зацикливаний [145], выхода из тупиков U-образной формы [146], движения в лабиринтах [147], узких проходах [148], движения вдоль криволинейных траекторий [149].

Как правило, системы управления движением используют алгоритмы, основанные на нечеткой логике в составе гибридный систем.

1.3.3. Планирование пути с применением нейронных сетей

Среди методов машинного обучения наиболее распространены сегодня методы планирования, строящиеся на нейронных сетях глубокого обучения [150, 151]. За счет хороших аппроксимирующих свойств и возможностей по классификации нейронные сети применяются в неопределенных средах.

Распространенная схема применения нейронной сети – ее использование для параметрической настройки базового алгоритма планирования. Такой подход применен в работе [152], в которой рассматривается нейросетевой алгоритм планирования группового движения подвижных объектов в частично некартографированной среде. Нейронная сеть применяется для

настройки параметров алгоритмов, используемых для планирования движения группы, которая должна сохранять при движении заданные заранее строй. При этом движение осуществляется в двумерной среде.

Работа [153] посвящена задаче планирования траекторий движения группы роботов, которые функционируют в неопределенной среде. Каждый робот реализует нейронную сеть, совокупность которых планирует траектории движения, основываясь на обмене данными между соседними подвижными объектами. Для подтверждения эффективности разработанных алгоритмов проведено численное моделирование разведки группой некоторой области, в которой произошло природное бедствие. Также в данной работе приведены результаты натурных экспериментов.

В статье [154] предложена нейронная сеть глубокого обучения, решающая задачу планирования пути группы мобильных объектов, осуществляющих прогнозирование положения и преследование убегающего объекта. При этом в нейронной сети учитываются ограничения и запаздывания в каналах связи. Нейронная сеть функционирует успешно, что подтверждается результатами моделирования.

Разработанная в работе [155] глубокая нейронная сеть решает задачи распознавания и сопровождения объектов интереса, а также планирование движений элементов группы роботов, осуществляющей обнаружение и слежение за объектами интереса. В статье приводятся результаты проведенных численных и натурных экспериментов, которые подтверждают эффективность предложенный нейросетевых решений.

В статье [156] используется две нейронных сети, одна из которых осуществляет классификацию состояния робота по пяти классам. Вторая нейронная сеть осуществляет обнаружение некорректных ситуаций при планировании действий складского робота. Обе нейронные сети объединены в общую систему посредством конечного автомата. Авторами статьи

приводятся результаты численных и натурных экспериментов, которые подтверждают работоспособность разработанных сетей.

В статье [157] используется рекуррентная нейронная сеть, осуществляющая планирование движения группы подвижных объектов, которые должны двигаться в заданной области, заполняя оптимально по заданному критерию заранее заданную область. При этом задача формирования строя вначале сформулирована в виде оптимизационной задачи по негладкому критерию. Затем сформулированная задача решается рекуррентной нейронной сетью. Система планирования движения и удержания заданной конфигурации исследована в данной статье с помощью численного моделирования и путем проведения реальных экспериментов, которые подтверждают успешное функционирование нейронной сети.

Работа [158] посвящена разработке системы управления подвижным объектом колесного типа в двумерной среде с частично некартографированной среде. Изучается специальный вопрос конструирования структуры глубокой нейронной сети, которая применяется для планирования направления движения подвижного объекта. В результате предложена каскадная нейронная сеть, каждый следующий каскад которой вложен в предыдущий каскад. Указанная структура позволяет заменить один сложный нейросетевой классификатор, на каскад более простых нейронный сетей, осуществляющих классификацию текущей ситуации на 2 класса. Работоспособность и преимущества представленного подхода подтверждается результатами численных и натурных экспериментов.

В статье [159] решается задача обнаружения препятствий и планирования движения в двумерной некартографированной среде. Обе перечисленные задачи решаются с применением сетей глубокого обучения. Первая сеть осуществляет детектирование свободных от препятствий секторов, а вторая – планирует траекторию, обеспечивающую обход возникающих одиночных препятствий. Следует отметить, что нейронные сети в данной

работе функционируют совместно с глобальным алгоритмом планирования, использующем априорную информацию о среде.

Статья [160] посвящена решению известной задачи определения положения и картографирования (SLAM) на базе глубокой нейронной сети. При этом дополнительно, нейронная сеть планирует траекторию движения и осуществляет компенсацию ошибок одометрии, которые, как известно, могут накапливаться. Для решения оставленной задачи синтезирована и обучена нейронная сеть.

Нейронные сети в динамических системах не гарантируют асимптотически устойчивых решений, поэтому для решения указанной проблемы часто применяются составные системы. Например, в статье [161] разработана нейронная сеть, которая функционирует совместно с алгоритмом потенциальных полей. Такое сочетание алгоритмов позволяет решать проблему локальных минимумов и получать гладкие траектории.

В работе [162] предлагается совместное применение нейронной сети и нечеткого алгоритма для планирования группового движения в двумерной среде с препятствиями, положение которых заранее неизвестно. Роботы и препятствия представлены материальными точками. В статье показаны преимущества и более высокие показатели эффективности разработанной составной системы.

Статья [163] также содержит подход, заключающийся в сочетании нейронных сетей с нечеткими правилами планирования траекторий подвижных объектов в двумерных средах. Базовым алгоритмом является набор нечетких правил, а нейронная сеть используется для их настройки. Представленное решение позволяет оптимизировать число активируемых нечетких правил.

В связи с высокой вычислительной сложностью нейронных сетей и сложностью процедур их обучения большое число работ посвящено оптимизации структуры и повышению эффективности обучения. В частности,

статья [164] развивает метод обучения глубокой нейронной сети, используемой для выбора направления движения подвижного объекта. Предложенные метод позволяет изменять число слоев нейронной сети, которое используется для решения указанной задачи, в зависимости от текущей ситуации. При этом число используемых слоев получается автоматически в ходе поэтапной обработки данных нейронной сетью.

В работах [165 – 167, 196] предложена итерационная процедура формирования обучающей выборки, которая положена в основу одного из методов, разработанных в данной диссертации. Недостатком предложенного в указанных работах метода является то, что в нем в обучающую выборку добавлялись только те примеры, в которых возникают коллизии. Однако причины коллизий могут быть заключены в ранее неверно принятых решениях.

Статья [170] посвящена повышению вычислительной эффективности глубоких нейронных сетей за счет динамического изменения порогового входа нейронов. Высокая производительность такой сети продемонстрирована на примере решения наивгационной задачи.

На основе сделанного обзора можно сделать вывод о том, что применение нейронных сетей для решения различных задач становится все более популярным. Некоторые ограничения накладывают сложности с гарантией асимптотической устойчивости при применении нейронных сетей для управления динамическими системами. Тем не менее сегодня нейронные сети применяются в системах управления наземных колесных и гусеничных подвижных объектов [169], морских платформ [168, 173], шагающих роботов [171], манипуляторов [172] и беспилотных летательных аппаратов [174].

Основные проблемы разработки нейросетевых систем – автоматизация и снижение трудоемкости процессов создания обучающих выборок,

синтез структуры нейронной сети, повышение вычислительной эффективности нейросетевых планировщиков.

1.3.4. Планирование пути с применением биоинспирированных эвристик

К наиболее часто используемым биоинспирированным эвристикам относятся муравьиные и пчелиные алгоритмы, роевые алгоритмы, а также различные стайные механизмы.

Муравьиный алгоритм – это алгоритм роевого интеллекта, разработанный Марко Дориго в 1992 году в его докторской диссертации [175]. Данный подход относится к типу популяционных и используется для нахождения глобального оптимума. В системах планирования пути алгоритм оптимизации муравьиной колонии используется в связи с тем, что он основан на поведении муравьев и их способности находить кратчайший путь от их гнезда к источнику пищи.

В работе [176] представлено исследование, посвященное применению метода оптимизации муравьиной колонии для планирования траекторий мобильных роботов в реальном времени. Благодаря применению муравьиного алгоритма авторам указанной статьи удалось увеличить скорость сходимости, вариативность решения и вычислительную эффективность с генетическим алгоритмом.

В работе [177] разработано несколько модификаций муравьиного алгоритма. Рассмотрены две задачи – коммивояжера и маршрутизации транспорта. Представленные в статье результаты позволяют оптимизировать решение указанных выше задач. При этом проведено сравнительное экспериментальное исследование, показавшее эффективность предложенных муравьиных алгоритмов по сравнению с классическим муравьиным алгоритмом, генетическим алгоритмом и методов границ и ветвей.

Различные роевые и стайные алгоритмы управления в трехмерных средах исследованы в работах [9, 178 – 180]. В докладе [178] предложен алгоритм оптимизации частиц роя. Данный алгоритм обладает низкой вычислительной стоимостью при решении задач группового движения. В работе [179] исследован голубиный алгоритм для решения задачи навигации. Данный алгоритм использован для планирования траекторий движения группы БЛА. Статья [180] посвящена решению задачи планирования пути в неопределенной динамической среде с использованием алгоритма серых волков. Предложенный алгоритм позволяет учитывать расстояние между роботами, дистанцию до целевой точки, расстояния от роботов до препятствий.

1.4. Выводы по главе 1

Проведенный анализ позволяет сделать вывод, что в настоящее время типичными вопросами планирования траекторий являются:

1) Учет ограничений с точки зрения формирования физически реализуемых траекторий, учитывающих динамические свойства роботов.

2) Оптимизация вычислительной сложности и сходимость алгоритмов к оптимальным траекториям.

3) Учет неопределенности, связанной с отсутствием полной карты окружающей среды и оптимизацией процессов планирования траекторий в динамических средах.

Актуальными проблемами применения нейронных сетей в задача планирования траекторий, основными направлениями исследований являются:

1) Снижение трудоемкости синтеза оптимизированной структуры глубокой нейронной сети, соответствующей решаемой проблеме;

2) Автоматизация процесса обучения и формирования обучающей выборки для глубокой нейронной сети.

Указанные проблемы исследуются в данной диссертации.

Глава 2. Двухэтапный метод планирования глобальной траектории движения робота

2.1. Постановка задачи

Рассматривается плоская среда с препятствиями прямоугольной формы [197, 199]. Препятствия определены координатами левого нижнего угла $p_{pi}(x_{pi}, y_{pi})$ и размерами r_{pi} вдоль оси OX_g и b_{pi} вдоль оси OY_g. Препятствия могут образовывать различные конфигурации, как представлено на рис. 2.1. Ориентация и положение робота заданы углом φ и координатами (x, y), определяющим направление движения в неподвижной системе координат OX_gY_g. Цель определяется координатами (x_t, y_t).

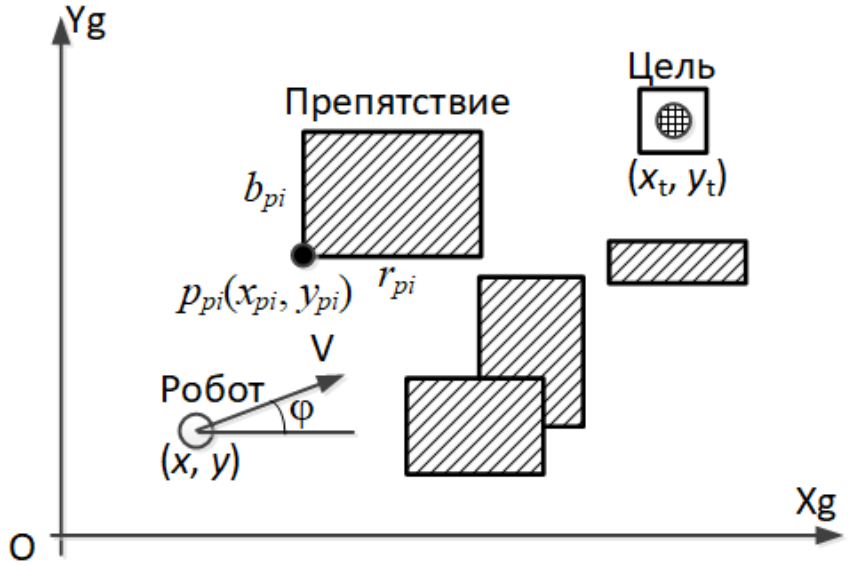


Рисунок 2.1 – Координаты робота, препятствия и цели

Учитывая, что боковое скольжение отсутствует, а скорость объекта направлена вдоль оси OX связанной системы координат, модель движения мобильного робота описывается следующими уравнениями [181]:

$$\dot{x} = V\cos\varphi, \tag{2.1}$$

$$\dot{y} = V\sin\varphi, \tag{2.2}$$

$$\dot{\varphi} = \omega, \tag{2.3}$$

$$\dot{V} = -\alpha_v(V - V^*), \tag{2.4}$$

$$\dot{\omega} = -\alpha_{\omega}(\omega - \omega^{*}), \quad (2.5)$$

где *x*, *y* – координаты робота в неподвижной системе; φ – угол рысканья; V – линейная скорость робота; ω – угловая скорость робота; α_v, α_φ – коэффициенты, определяющие динамику объекта; V* – задание по линейной скорости;, φ* – желаемый угол поворота.

Требуется разработать алгоритм планирования, обеспечивающий учет следующих ограничений, обусловленных свойствами и текущим состоянием робота и среды:

1) Должна учитываться зависимость радиуса поворота робота от текущей скорости движения, т.е. должен быть физически реализуем;

2) Должно быть найдено компромиссное решение, учитывающее степень оптимальности траектории и вычислительные затраты на нахождение этой траектории;

3) Должна учитываться зависимость скорости движения от расстояния до препятствий;

4) Должна быть обеспечена возможность учета препятствий, не обозначенных на карте и обнаруженных системой технического зрения;

5) Должна обеспечиваться гладкость планируемой траектории;

6) Необходимым условием построения траектории являются условия ее прохождения через начальное положение и робота и целевую точку. Скорость робота в целевой точке равна нулю. Требования к угловой ориентации робота в целевой точке не предъявляются.

Чтобы показать необходимость учета текущей скорости движения робота при планировании пути, проведем моделирование уравнений (2.1) – (2.5).

Пусть желаемый угол движения определяется как арктангенс угла наклона прямой, связывающей текущее и целевое положение робота

$$\varphi^{*} = \operatorname{atan}\frac{y_t - y}{x_t - x}, \quad (2.6)$$

где (x_t, y_t) – координаты целевой точки.

Данный желаемый угол движения позволяет сформировать уставку по угловой частоте вращения робота

$$\omega^* = k_\omega(\varphi^* - \varphi), \tag{2.7}$$

где k_ω – коэффициент пропорциональности.

В случае, если уставка по желаемой угловой частота вращения ω* превышаем максимально возможное значение, то она ограничивается дополнительным условием

$$\begin{aligned} \omega^* &= k_\omega(\varphi^* - \varphi), \quad \forall|\omega^*| \le \omega_{max}, \\ \omega^* &= \omega_{max} sign\big(k_\omega(\varphi^* - \varphi)\big), \quad \forall|\omega^*| > \omega_{max}. \end{aligned} \tag{2.8}$$

Скорость движения робота определяется следующим образом

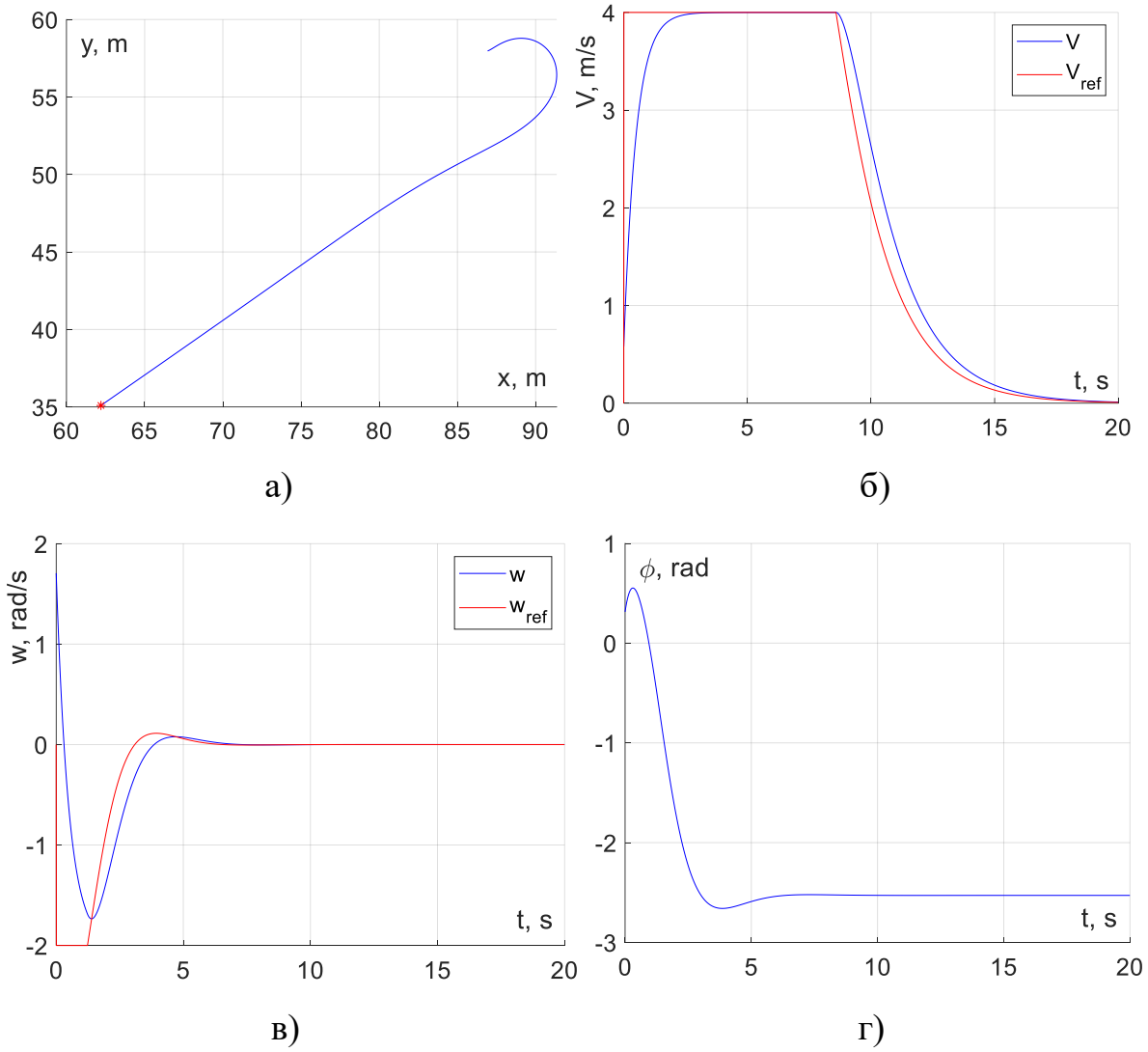
$$\begin{aligned} V^* &= V_{max}\frac{r}{r_s}, \quad \forall\ V^* \le V_{max}, \\ V^* &= V_{max}, \quad \forall\ V^* > V_{max}, \end{aligned} \tag{2.9}$$

где V_{max} – максимальная (или крейсерская) скорость движения робота; r – расстояние до целевой точки; r_s – длина тормозного пути робота.

Пример траектории движения робота в заданную точку представлен на рис. 2.2, из которого видно, что робот осуществляет успешный маневр и достигает целевой точки, показанной красной звездочкой. Отметим, что результаты моделирования получены в среде без препятствий. На рис. 2.2,а представлена траектория подвижного объекта в двумерной среде, на рис. 2.2,б – линейная скорость и ее задающее воздействие, на рис. 2.2,в – угловая скорость и ее задающее воздействие, а на рис. 2.2,г – изменение во времени угла ориентации робота.

Анализируя результаты моделирования, представленные на рис. 2.2, отметим, что направление поворота робота не задается системой управления, что принципиально важно в средах с препятствиями. Также отметим, что радиус разворота робота составляет примерно 5 – 6 метров, что может приводить к коллизиям при планировании траекторий. Также отметим, что

из-за отработки задающих воздействий с запаздыванием, реальная траектория робота всегда отличается от идеальной траектории. Таким образом, при планировании траектории движения необходимо учитывать инерционные свойства, текущее направление и скорость движения мобильного робота.



а) б)

в) г)

Рисунок 2.2 – Результаты моделирования движения робота

В этой связи рассмотрим систему управления мобильного робота, базирующуюся на карте, которая строится предварительно, но может уточняться на основе данных системы технического зрения.

Предполагается, что мобильный робот оснащен системой технического зрения (СТЗ), представляющей собой сектор окружности радиусом

$R_{СТЗ}$ и углом $\varphi_{СТЗ}$. Сектор системы технического зрения робота представлен на рис. 2.3.

Мобильный робот также оснащен системой глобальной навигации, которая позволяет определять собственное положение в неподвижной системе координат.

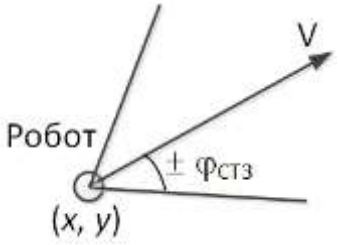


Рисунок 2.3 – Сектор системы технического зрения робота

Препятствия могут быть стационарными или перемещаться. Их скорости не измеряются роботом, однако могут оцениваться по результатам измерения положений. Максимальная скорость препятствий равна V_{obs_max}.

Препятствия также разделяются на априорно известные, нанесенные на карту и не нанесенные на карту. Последние обнаруживаются СТЗ робота, который может вносить их на карту. Все подвижные препятствия на априорной карте отсутствуют. Неподвижные препятствия могут быть нанесены на карте или отсутствовать.

Требуется разработать систему планирования и управления движением робота, решающую следующие задачи:

– планирование траектории движения робота, обеспечивающей движение в целевую точку без коллизий;

– планирование скорости и направления движения робота V*, φ* при движении в целевую точку (x_t, y_t);

– учет ограничений на угол поворота робота, в зависимости от текущей скорости и инерционных свойств робота, описываемого уравнениями (1) – (5).

2.2. Алгоритм нанесения априорно неизвестных препятствий на карту по данным СТЗ

Первичные алгоритмы обработки данных от СТЗ и алгоритмы распознавания препятствий в данном разделе не рассматриваются. Считается, если препятствие попало в область действия СТЗ, то оно обнаружено. В данном разделе рассматривается процесс построения препятствия на карте по видимой его части.

Пусть СТЗ обнаружило линию длиной a_{obs}. В этом случае на карту заносится квадрат со стороной a_{obs}, как показано на рис. 2.4.

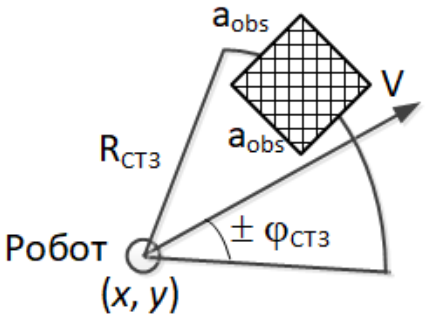


Рисунок 2.4 – Нанесение препятствия на карту при обнаружении линии

Если СТЗ обнаружила угол, больше развернутого (больше 180 °), то на карту наносится препятствие, представляющее собой параллелепипед со сторонами a_{obs} и b_{obs}, как показано на рис. 2.5.

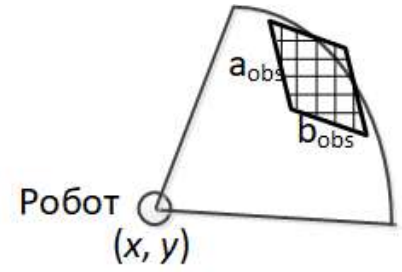


Рисунок 2.5 – Нанесение препятствия на карту при обнаружении угла больше 180 °

При обнаружении угла, меньше 180 °, на карту наносится препятствие, показанное на рис. 2.5.

Как можно заметить из рис. 2.4 – 2.6, невидимая часть препятствия строится таких же размеров, что и видимая часть. Данный принцип используется и при обнаружении более сложных конфигураций, которые получаются путем комбинирования углов.

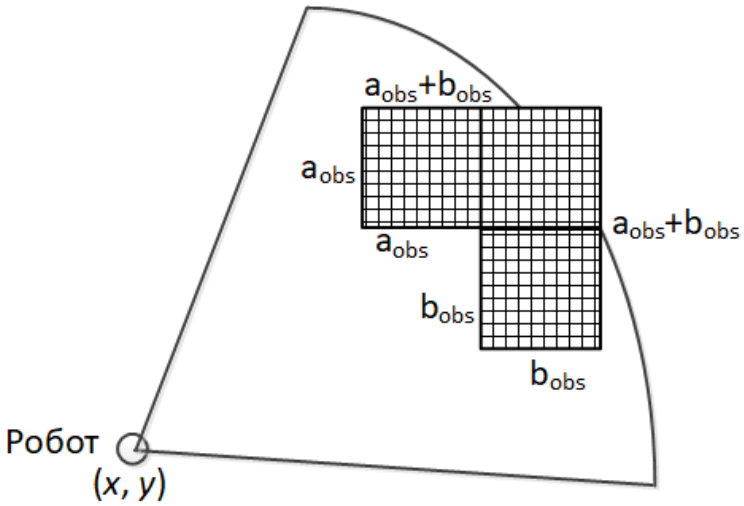


Рисунок 2.6 – Нанесение препятствия на карту при обнаружении угла меньше 180 °

На рис. 2.7 представлен пример построения невидимой части препятствий для обрыва. При этом имеется ввиду, что сложные формы препятствий могут быть получены в результате последовательности наблюдений.

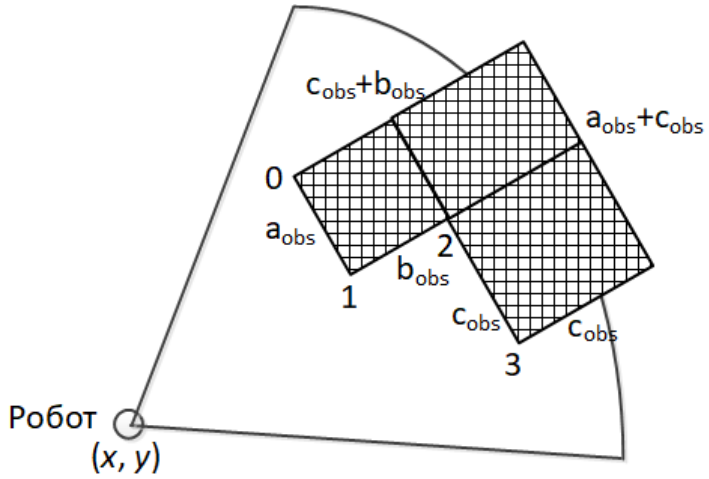


Рисунок 2.7 – Нанесение препятствия на карту при обнаружении обрыва

На рис. 2.7 ломаная 0 – 1 – 2 – 3 является видимой частью. Остальная часть препятствия достраивается автоматически.

2.3. Алгоритм планирования траектории и управления движением мобильного робота

2.3.1 Синтез алгоритма изменения скорости движения

В общем случае, скорость движения робота является функцией расстояния до цели, расстояния до препятствий и разности между текущим и желаемым направлением движения робота. Такая зависимость обусловлена тем фактом, что робот при приближении к препятствиям и при поворотах должен уменьшать свою скорость. Введем коэффициенты, снижающие скорость робота при его повороте и при приближении к препятствиям:

$$V^* = V_{max}\frac{r}{r_s}k_\varphi k_{obs}, \quad \forall\, V^* \leq V_{max}, \qquad V^* = V_{max}, \quad \forall\, V^* > V_{max,} \tag{2.10}$$

где k_φ – коэффициент, зависящий от разности между текущим и желаемым направлением движения робота; k_{obs} – коэффициент, зависящий от расстояния до препятствия.

Коэффициент k_φ, определяющий снижение скорости в зависимости от ткущего направления движения, определяется в соответствии с графиком, представленным на рис. 2.8.

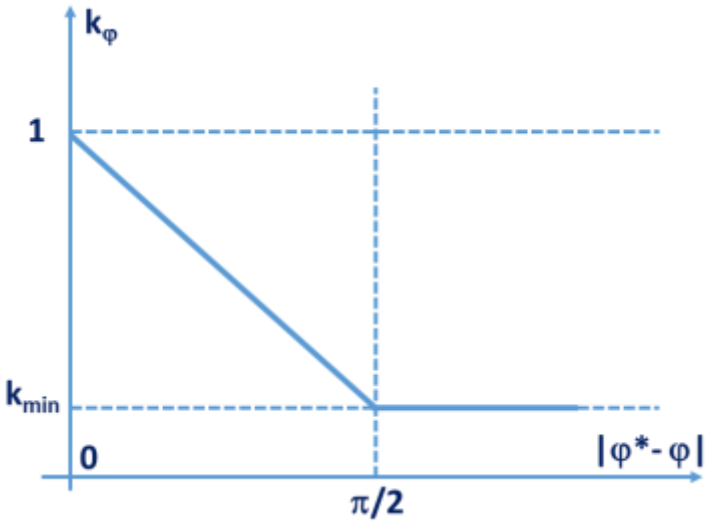


Рисунок 2.8 – Зависимость коэффициента k_φ от |φ*-φ|

Согласно указанной зависимости, если направление движения робота совпадает с желаемым направлением движения, то от движется со скоростью V_{max}. При отклонении текущего направления от желаемого направления движения скорость снижается линейно. Тогда, согласно рис. 2.8 получим

$$k_\varphi = 1 + (k_{min} - 1)\frac{|\varphi* - \varphi|}{\frac{\pi}{2}}, \quad \forall|\varphi* - \varphi| < \pi/2,$$
$$k_\varphi = k_{min}, \quad \forall|\varphi* - \varphi| \geq \pi/2. \tag{2.11}$$

Величина коэффициента k_{min} зависит от конкретных динамических характеристик робота, примем эту величину равной $k_{min} = 0.05$, т.е. при углах поворота более 90 градусов скорость робота уменьшается до 5 % от максимального значения.

Рассмотрим теперь зависимость коэффициента k_{obs} от расстояния до препятствий. Здесь требуется выделить две ситуации. Первая ситуация состоит в том, что обнаруженное препятствие не мешает движению робота к целевой точке. Данная ситуация представлена на рис. 2.9, на котором робот движется к цели через сектор I, а препятствие занимает сектор II.

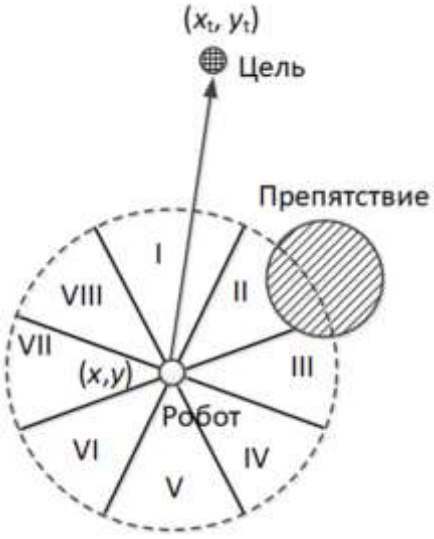


Рисунок 2.9 – Препятствие не мешает движению к цели

Для этого случая требуется снижение скорости робота, осуществляемое за счет изменения коэффициента k_{obs} в соответствии с графиком k_{obs1}, который представлен на рис. 2.10.

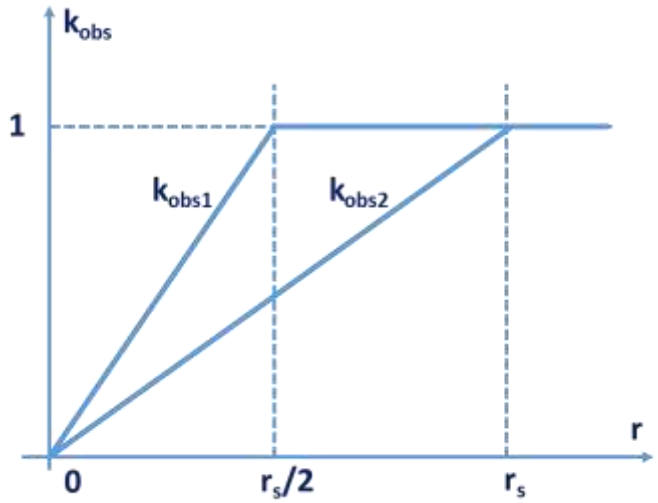


Рисунок 2.10 – Графики зависимости коэффициента k_{obs} от расстояния до препятствия r

В соответствии с графиками, представленными на рис. 2.10, получаем

$$k_{obs1} = 2\frac{r_{obs}}{r_s}, \quad \forall r_{obs} < 0.5r_s, \qquad k_{obs1} = 1, \quad \forall r_{obs} \geq 0.5r_s, \tag{2.12}$$

где r_{obs} – расстояние от робота до препятствия.

Рассмотрим теперь второй случай, когда препятствие расположено так, что оно мешает движению робота к цели как, например, на рис. 2.11.

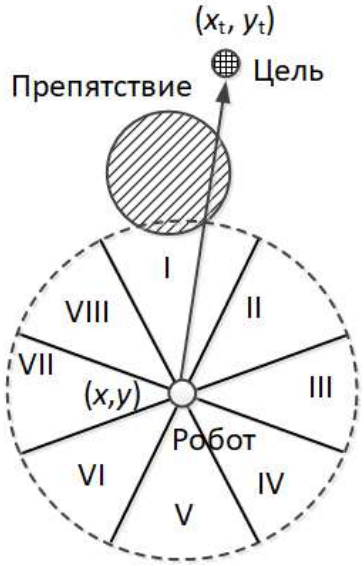


Рисунок 2.11 – Препятствие мешает движению к цели

В этом случае изменение коэффициента k_{obs} производится в соответствии с графиком k_{obs2}, который представлен на рис. 2.10. В соответствии с данным графиком получаем

$$k_{obs2} = \frac{r_{obs}}{r_s}, \quad \forall r_{obs} < r_s,$$
$$k_{obs1} = 1, \quad \forall r_{obs} \geq r_s. \tag{2.13}$$

Таким образом, согласно рис. 2.10 и выражениям (2.12), (2.13), при появлении препятствия по ходу движения робот начинает уменьшать скорость раньше, чем при появлении препятствий в стороне от траектории его движения.

Выражения (2.10) – (2.13) определяют закон изменения линейной скорости движения робота в среде с препятствиями. Входы и выходы блока вычисления желаемой скорости движения V* представлены на рис. 2.12.

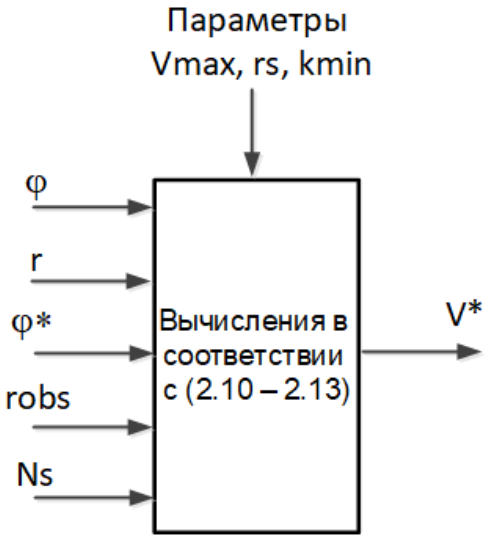


Рисунок 2.12 – Блок планирования линейной скорости движения

Входами блока планирования линейной скорости движения являются: текущий угол рысканья робота φ, желаемый угол рысканья φ*, расстояние до цели r, расстояние до препятствия r_{obs} и номер сектора N_s, в котором расположено препятствие. Параметрами блока являются крейсерская скорость робота V_{max}, коэффициент k_{min}, определяющий минимальную скорость движения при повороте на 90 градусов, и безопасная дистанция r_s, в качестве которой может быть взят тормозной путь.

2.3.2 Синтез алгоритма изменения направления движения

Задача планирования желаемого направления движения намного сложнее, чем задача планирования скорости движения [4]. В качестве алгоритмов планирования обычно используются глобальный планировщик, основанный на вычислении некоторого оптимального пути на известной карте. Однако в таком алгоритме планирования сложно учесть динамические свойства объекта и неопределенность среды, поэтому он дополняется локальным алгоритмом, корректирующим путь и обеспечивающим его гладкость.

Наиболее часто алгоритмы поиска пути состоят из двух частей. Первая часть осуществляет поиск пути в дискретном или непрерывном конфигурационном пространстве, которое представляет собой пространство функционирования работа, включающее разрешенные и запрещенные состояния [44]. Вторая часть осуществляет сглаживание полученного пути, обеспечение его безопасности и физическую реализуемость.

В данной главе будем синтезировать такой двухэтапный алгоритм планирования траектории движения. При этом, в соответствии с результатами анализа, выполненными в главе 1, в качестве базового планировщика движения будем использовать субоптимальный алгоритм случайных исследующих деревьев (RRT) [95] и его модификации [105, 107, 182, 183].

Будем представлять карту среды в виде матрицы, на которой 0 – свободная ячейка, 1 – текущее положение робота; 2 – целевое положение робота; 3 – ячейка занята препятствием. Размер ячейки примем равным l_c.

Для учета динамических ограничений введем следующее предположение. При минимальной (для танковых схем нулевой) скорости движения робот может реализовать поворот на любой угол в пределах ячейки, в которой он находится, т.е. он может переместиться в любую соседнюю ячейку. При максимальной скорости движения робот угол поворота робота ограни-

чен небольшим значением. Тогда будем полагать, что возможный физически реализуемый угол поворота экспоненциально уменьшается от ±180° до минимального угла $\Delta\varphi_{min}$, при повышении скорости от V_{min} до максимального значения V_{max}. Такая зависимость допустимого угла поворота от линейной скорости может быть описана следующим выражением

$$|\varphi^* - \varphi|_{доп} = \Delta\varphi_{min} + (\pi - \Delta\varphi_{min}) e^{-\frac{V - V_{min}}{\alpha(V_{max} - V_{min})}}, \qquad (2.14)$$

где α – постоянный параметр.

Внешний вид зависимости (2.14) при $\Delta\varphi_{min}$=0.1, V_{min}=0.5, V_{max}=5, α=0.2 представлен на рис. 2.13.

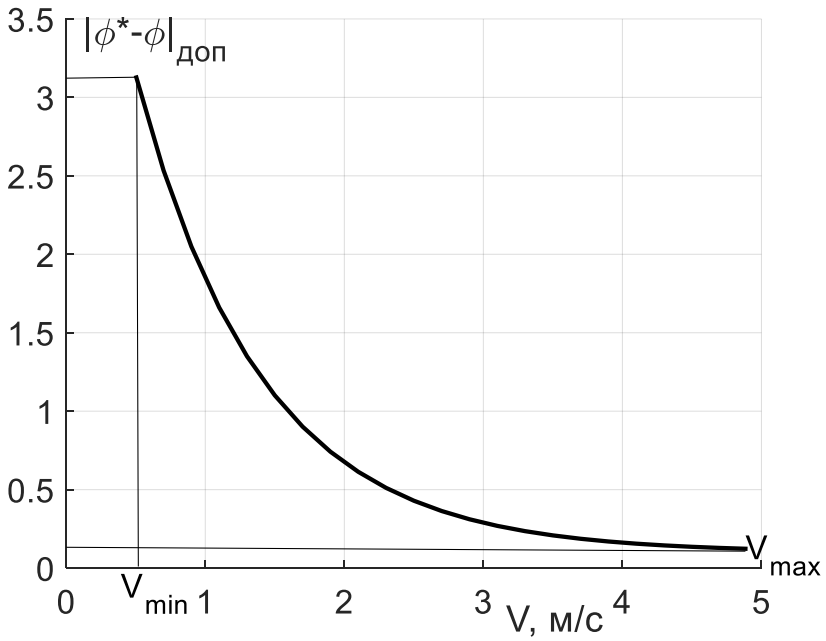


Рисунок 2.13 – Вид зависимости допустимого угла поворота от скорости движения согласно (2.14)

В данной диссертации предлагается учитывать указанной ограничение, введя на карту виртуальные препятствия, которые будут перекрывать физически нереализуемые сектора движения. Это позволяет использовать произвольные алгоритмы планирования, ориентированные на использование карты среды.

Процедура введения виртуальных препятствий поясняется на рис. 2.14.

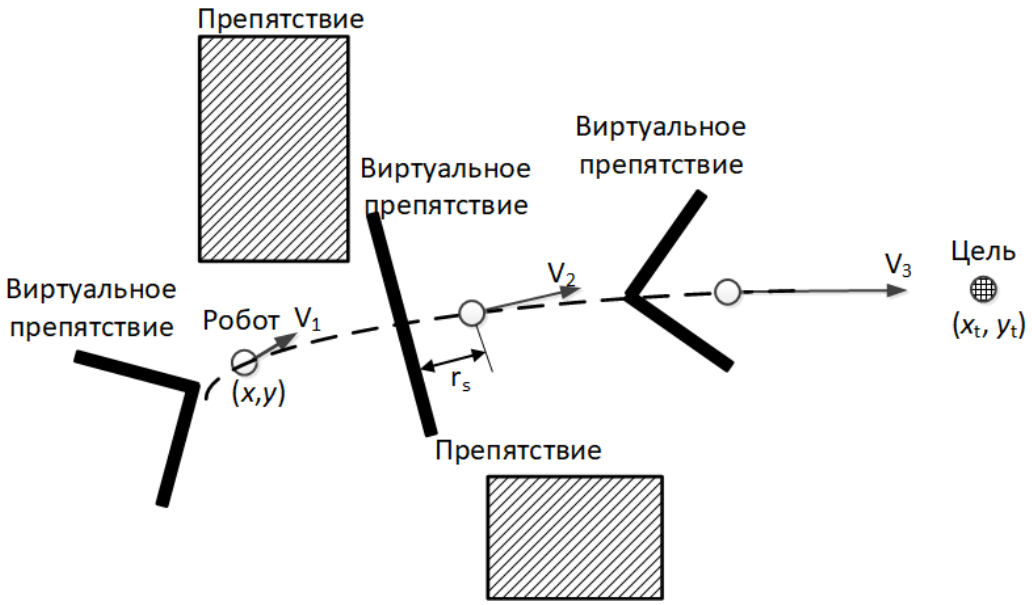


Рисунок 2.14 – Введение виртуальных препятствий

Согласно рис. 2.14 вводимое виртуальной препятствие располагается позади робота на расстоянии r_s и имеет V-образную форму. При этом угол положения ребер препятствия зависит от текущей скорости. На рис. 2.15 показан случай, когда $V_3 > V_2 > V_1$. На низких скоростях ребра данного препятствия расположены под выпуклым углом по отношению к роботу. При увеличении скорости данный угол становится развернутым, а затем острым.

В общем случае, возможно определение данного угла в виде непрерывной функции скорости робота, в соответствии с графиком, представленным на рис. 2.13. Однако при наличии неопределенности в характеристике, представленной на рис. 2.13, более целесообразно вводить дискретно задаваемые виртуальные препятствия. В данной работе предлагается использование четырех типов виртуальных препятствий, соответствующих движению на малых скоростях $V < V_1$, движению на средних скоростях, ограниченных некоторым интервалом $V \in [V_1\ V_1]$, движению на крейсерских скоростях, ограниченных интервалом $V \in (V_2\ V_3]$, и на высоких скоростях $V > V_3$.

Для движения на скоростях $V < V_1$ виртуальное препятствие отсутствует, для скорости $V \in [V_1\ V_2]$ виртуальное препятствие имеет V-образную

форму и расположено под выпуклым углом по отношению к роботу. При скорости $V \in (V_2\ V_3]$ виртуальное препятствие представляет собой прямую линию (препятствие V-образной формы с развернутым углом), расположенную за роботом. Наконец, при $V > V_3$ препятствие расположено позади робота острым углом к нему.

Опишем процедуру добавления виртуальных препятствий в карту среды. Отметим, что матрица, описывающая среду движения робота, формируется в соответствии рис. 2.15.

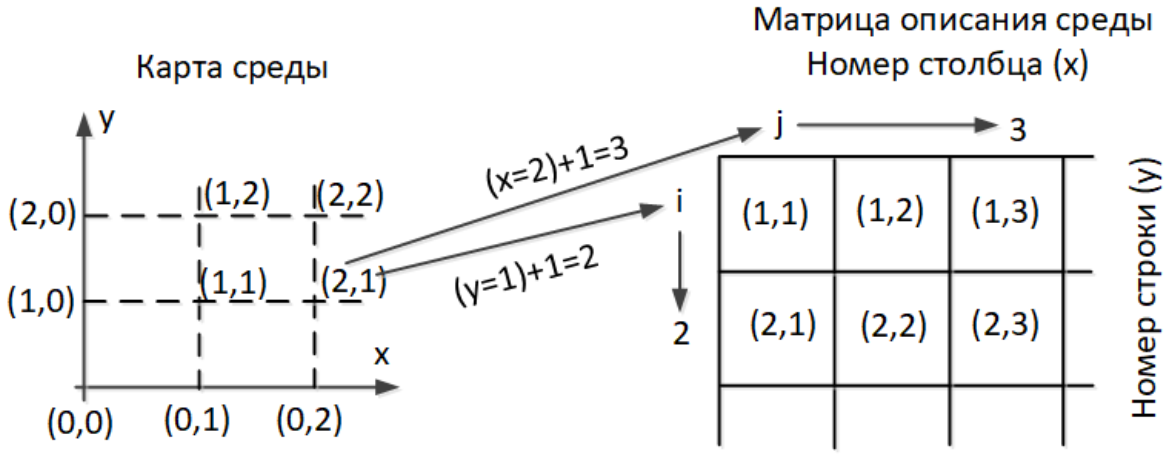


Рисунок 2.15 – Особенности преобразования карты среды в матрицу

Согласно рис. 2.15, карта среды разбита на квадратные ячейки заданного размера. Каждой ячейке с координатами (x,y) ставится в соответствие элемент матрицы Map (j,i), причем j = round(x+1), i=round(y+1). Тогда положение робота и целевой точки записываются в матрицу среды следующим образом:

$$Map(round(y+1), round(x+1)) = 1, \qquad (2.15)$$

$$Map(round(y_t+1), round(x_t+1)) = 2. \qquad (2.16)$$

Препятствия состоят из фрагментов прямоугольной формы задаются координатами (x_{obs}, y_{obs}) нижнего левого угла, а также размером r_p по оси Oy и b_p по оси Ox.

В этом случае запись координат препятствия в матрицу, описывающую среду функционирования робота, осуществляется в соответствии с выражением

$$Map(i,j) = 3, \quad (2.17)$$
$$i = \overline{round(y_{obs}) + 1, round(y_{obs}) + 1 + r_p},$$
$$j = \overline{round(x_{obs}) + 1, round(x_{obs}) + 1 + b_p}.$$

Т.к. при $V < V_1$ виртуальное препятствие отсутствует, то вначале рассмотрим случай, когда скорость робота находится в диапазоне $V \in (V_1\ V_2]$. Для этого случая виртуальное препятствие расположено за роботом на расстоянии r_s по оси, ориентированной вдоль направления его движения. Для данного диапазона скоростей рассмотрим случаи, когда робот движется в различные ячейки.

Будем считать, что робот движется в ячейку I (рис. 2.16а), если направление его движения ограничено величиной φ_1, равной

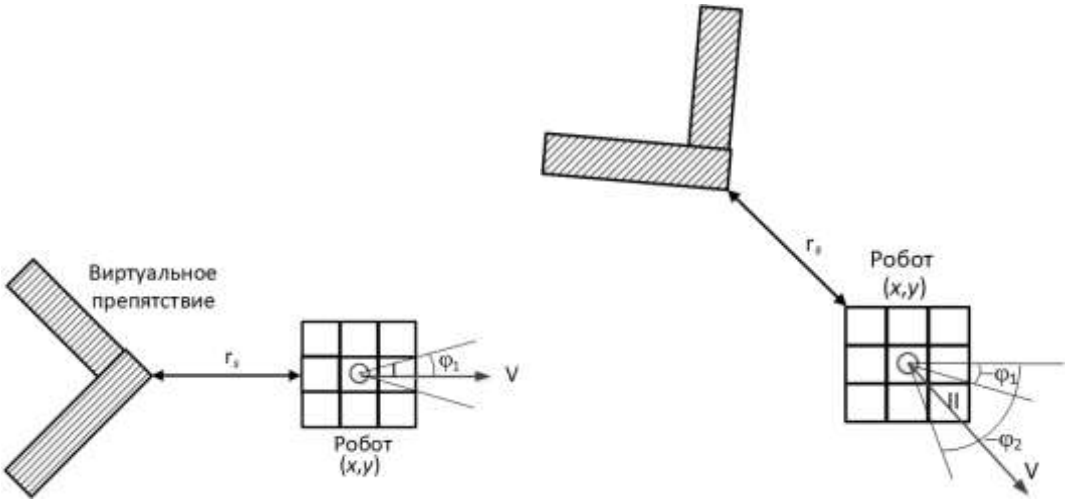
$$|\varphi| \le \varphi_1 = \operatorname{atan}\frac{0.5}{1.5} = 0.32. \quad (2.18)$$

Тогда согласно рис. 2.17а получаем, что координаты ячеек виртуального препятствия определяются выражениями

$$x_{obsv1} = x - r_s,\ y_{obsv1} = y, x_{obsv1} = x - r_s,\ y_{obsv1} = y - 1, \quad (2.19)$$
$$x_{obsv1} = x - r_s - 2,\ y_{obsv1} = y - 2, x_{obsv1} = x - r_s - 3,\ y_{obsv1} = y - 3,$$
$$x_{obsv1} = x - r_s - 4,\ y_{obsv1} = y - 4, x_{obsv1} = x - r_s - 1,\ y_{obsv1} = y + 1,$$
$$x_{obsv1} = x - r_s - 2,\ y_{obsv1} = y + 2, x_{obsv1} = x - r_s - 3,\ y_{obsv1} = y + 3,$$
$$x_{obsv1} = x - r_s - 4,\ y_{obsv1} = y + 4,$$

Аналогично, будем считать, что робот движется в ячейку II (рис. 2.16б), если направление его движения ограничено сектором от $-\varphi_1$ до $-\varphi_2$, причем угол φ_2 определяется следующим выражением

$$-\varphi_2 \le \varphi < -\varphi_1,\ \ \varphi_2 = \frac{\pi}{2} - 0.32 = 1.25. \quad (2.20)$$



а) Движение в ячейку I б) Движение в ячейку II

Рисунок 2.16 – Определение координат препятствия для случая $V \in (V_1\ V_2]$ при движении робота в ячейки I и II

В этом случае координаты виртуального препятствия определяются выражениями

$$x_{obsv1} = x - r_s,\ y_{obsv1} = y - r_s, x_{obsv1} = x - r_s,\ y_{obsv1} = y - r_s - 1, \quad (2.21)$$

$$x_{obsv1} = x - r_s,\ y_{obsv1} = y - r_s - 2, x_{obsv1} = x - r_s,\ y_{obsv1} = y - r_s - 3,$$

$$x_{obsv1} = x - r_s,\ y_{obsv1} = y - r_s - 4, x_{obsv1} = x - r_s - 1,\ y_{obsv1} = y - r_s,$$

$$x_{obsv1} = x - r_s - 2,\ y_{obsv1} = y - r_s, x_{obsv1} = x - r_s - 3,\ y_{obsv1} = y - r_s,$$

$$x_{obsv1} = x - r_s - 4,\ y_{obsv1} = y - r_s,$$

Аналогичным образом, получаем, что при движении робота в ячейку VIII (рис. 2.17а) направление движения и координаты виртуального препятствия определяются выражениями

$$\varphi_1 < \varphi \leq \varphi_2, \quad (2.22)$$

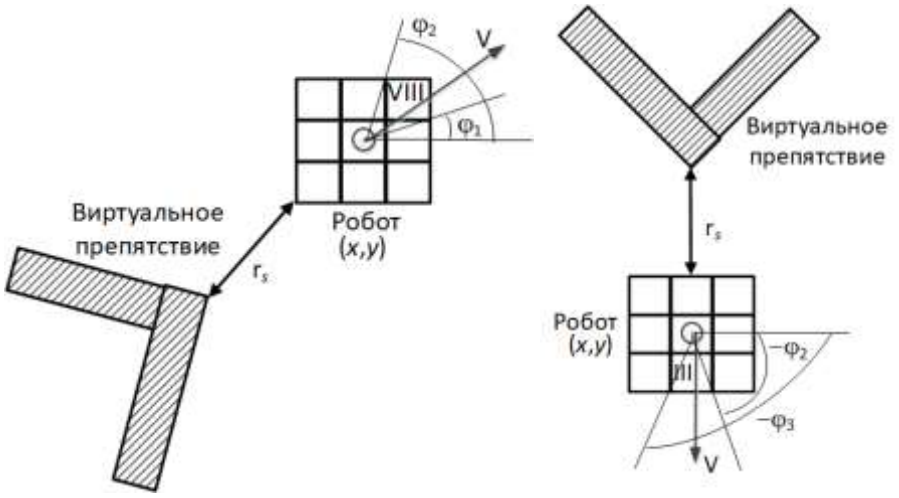
$$x_{obsv1} = x - r_s,\ y_{obsv1} = y + r_s, x_{obsv1} = x - r_s,\ y_{obsv1} = y - r_s + 1, \quad (2.23)$$

$$x_{obsv1} = x - r_s,\ y_{obsv1} = y - r_s + 2, x_{obsv1} = x - r_s,\ y_{obsv1} = y - r_s + 3,$$

$$x_{obsv1} = x - r_s,\ y_{obsv1} = y - r_s + 4, x_{obsv1} = x - r_s - 1,\ y_{obsv1} = y + r_s,$$

$$x_{obsv1} = x - r_s - 2,\ y_{obsv1} = y + r_s, x_{obsv1} = x - r_s - 3,\ y_{obsv1} = y + r_s,$$

$$x_{obsv1} = x - r_s - 4,\ y_{obsv1} = y + r_s,$$



а) Движение в ячейку VIII б) Движение в ячейку III

Рисунок 2.17 – Определение координат препятствия для случая $V \in (V_1\ V_2]$ при движении робота в ячейки VIII и III

Случай движения робота в ячейку III представлен на рис. 2.17б, на котором значение угла φ_3 определяется следующим выражением

$$\varphi_3 = \frac{\pi}{2} + 0.32 = 1.89. \quad (2.23)$$

При этом условие движения в ячейку III и координаты виртуального препятствия для данного случая определяются выражениями

$$-\varphi_3 \le \varphi < -\varphi_2, \quad (2.24)$$

$$x_{obsv1} = x,\ y_{obsv1} = y + r_s, x_{obsv1} = x - 1,\ y_{obsv1} = y + r_s + 1, \quad (2.25)$$

$$x_{obsv1} = x - 2,\ y_{obsv1} = y + r_s + 2, x_{obsv1} = x - 3,\ y_{obsv1} = y + r_s + 3,$$

$$x_{obsv1} = x - 4,\ y_{obsv1} = y + r_s + 4, x_{obsv1} = x + 1,\ y_{obsv1} = y + r_s + 1,$$

$$x_{obsv1} = x + 2,\ y_{obsv1} = y + r_s + 2, x_{obsv1} = x + 3,\ y_{obsv1} = y + r_s + 3,$$

$$x_{obsv1} = x + 4,\ y_{obsv1} = y + r_s + 4,$$

Условие движения в ячейку VII (рис. 2.18а) и координаты виртуального препятствия для этого случая определяются выражениями

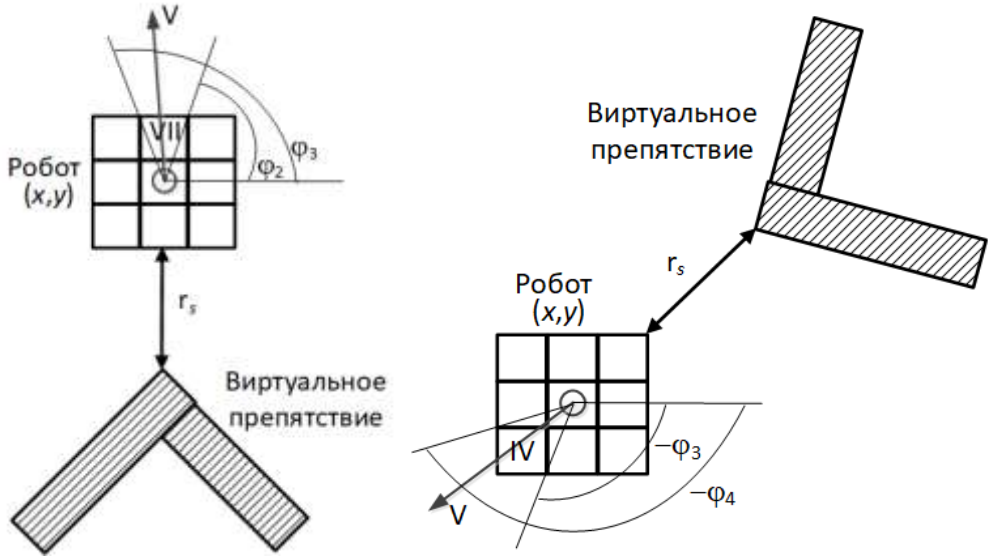
$$\varphi_2 < \varphi \le \varphi_3, \quad (2.25)$$

$$x_{obsv1} = x,\ y_{obsv1} = y - r_s, x_{obsv1} = x - 1,\ y_{obsv1} = y - r_s - 1, \quad (2.26)$$

$$x_{obsv1} = x - 2,\ y_{obsv1} = y - r_s - 2, x_{obsv1} = x - 3,\ y_{obsv1} = y - r_s - 3,$$

$$x_{obsv1} = x - 4,\ y_{obsv1} = y - r_s - 4, x_{obsv1} = x + 1,\ y_{obsv1} = y - r_s - 1,$$

$$x_{obsv1} = x + 2,\ y_{obsv1} = y - r_s - 2, x_{obsv1} = x + 3,\ y_{obsv1} = y - r_s - 3,$$
$$x_{obsv1} = x + 4,\ y_{obsv1} = y - r_s - 4.$$



а) Движение в ячейку VII б) Движение в ячейку IV

Рисунок 2.18 – Определение координат препятствия для случая $V \in (V_1\ V_2]$ при движении робота в ячейки VII и IV

Условие движения робота в ячейку IV (рис. 2.18б) и координаты виртуального препятствия для данного случая определяются выражениями

$$-\varphi_4 \le \varphi < -\varphi_3, \tag{2.27}$$

$$x_{obsv1} = x + r_s,\ y_{obsv1} = y + r_s, x_{obsv1} = x + r_s + 1,\ y_{obsv1} = y + r_s, \tag{2.28}$$
$$x_{obsv1} = x + r_s + 2,\ y_{obsv1} = y + r_s, x_{obsv1} = x + r_s + 3,\ y_{obsv1} = y + r_s,$$
$$x_{obsv1} = x + r_s + 4,\ y_{obsv1} = y + r_s, x_{obsv1} = x + r_s,\ y_{obsv1} = y + r_s + 1,$$
$$x_{obsv1} = x + r_s,\ y_{obsv1} = y + r_s + 2, x_{obsv1} = x + r_s,\ y_{obsv1} = y + r_s + 3,$$
$$x_{obsv1} = x + r_s,\ y_{obsv1} = y + r_s + 4.$$

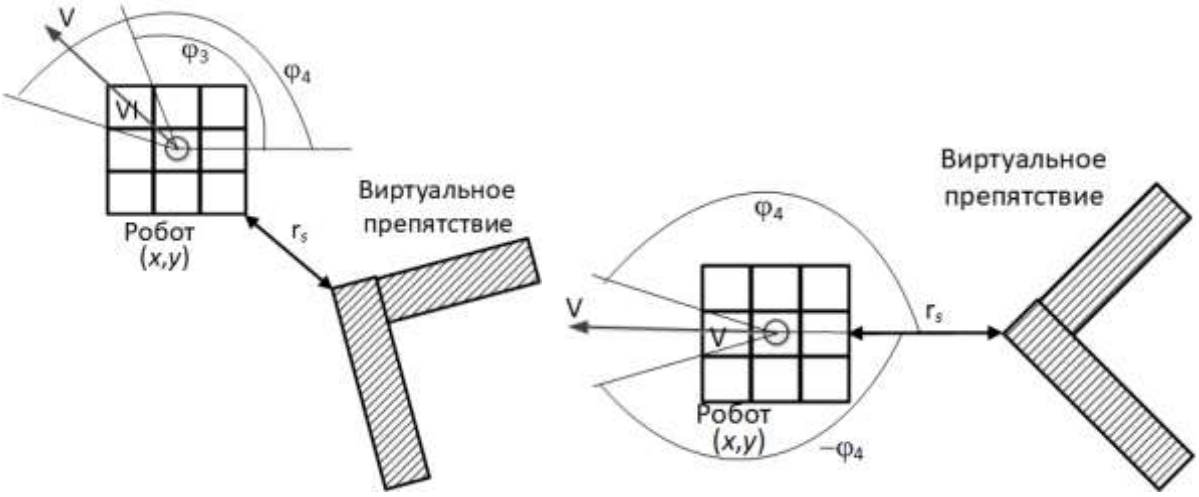
$$\varphi_4 = \pi - 0.32 = 2.82. \tag{2.29}$$

Условие движения в ячейку VI (рис. 2.19а) и координаты виртуального препятствия для этого случая определяются выражениями

$$\varphi_3 < \varphi \le \varphi_4, \tag{2.30}$$

$$x_{obsv1} = x + r_s,\ y_{obsv1} = y - r_s, x_{obsv1} = x + r_s + 1,\ y_{obsv1} = y - r_s, \tag{2.31}$$
$$x_{obsv1} = x + r_s + 2,\ y_{obsv1} = y - r_s, x_{obsv1} = x + r_s + 3,\ y_{obsv1} = y - r_s,$$

$$x_{obsv1} = x + r_s + 4,\ y_{obsv1} = y - r_s, x_{obsv1} = x + r_s,\ y_{obsv1} = y - r_s - 1,$$
$$x_{obsv1} = x + r_s,\ y_{obsv1} = y - r_s - 2, x_{obsv1} = x + r_s,\ y_{obsv1} = y - r_s - 3,$$
$$x_{obsv1} = x + r_s,\ y_{obsv1} = y - r_s - 4.$$



а) Движение в ячейку VI б) Движение в ячейку V

Рисунок 2.19 – Определение координат препятствия для случая V∈(V$_1$ V$_2$] при движении робота в ячейки VI и V

Условие движения в ячейку V (рис. 2.19б) и координаты виртуального препятствия для этого случая определяются выражениями

$$|\varphi| > \varphi_4, \tag{2.32}$$

$$x_{obsv1} = x + r_s,\ y_{obsv1} = y, x_{obsv1} = x + r_s + 1,\ y_{obsv1} = y - 1, \tag{2.33}$$
$$x_{obsv1} = x + r_s + 2,\ y_{obsv1} = y - 2, x_{obsv1} = x + r_s + 3,\ y_{obsv1} = y - 3,$$
$$x_{obsv1} = x + r_s + 4,\ y_{obsv1} = y - 4, x_{obsv1} = x + r_s + 1,\ y_{obsv1} = y + 1,$$
$$x_{obsv1} = x + r_s + 2,\ y_{obsv1} = y + 2, x_{obsv1} = x + r_s + 3,\ y_{obsv1} = y + 3,$$
$$x_{obsv1} = x + r_s + 4,\ y_{obsv1} = y + 4.$$

Таким образом, выражения (2.18) – (2.33) определяют условия движения робота в каждую из соседних ячеек и координаты виртуального препятствия для случая V∈(V$_1$ V$_2$].

Проводя аналогичные рассуждения для случая V∈(V$_2$ V$_3$], когда виртуальное препятствие расположено под развернутым углом к роботу (имеет форму прямой линии), получаем следующие выражения.

Для случая движения в сектор I ($|\varphi| \leq \varphi_1$):

$$x_{obsv1} = x - r_s,\ y_{obsv1} = y, x_{obsv1} = x - r_s,\ y_{obsv1} = y - 1, \quad (2.34)$$
$$x_{obsv1} = x - r_s,\ y_{obsv1} = y - 2, x_{obsv1} = x - r_s,\ y_{obsv1} = y - 3,$$
$$x_{obsv1} = x - r_s,\ y_{obsv1} = y - 4, x_{obsv1} = x - r_s,\ y_{obsv1} = y + 1,$$
$$x_{obsv1} = x - r_s,\ y_{obsv1} = y + 2, x_{obsv1} = x - r_s,\ y_{obsv1} = y + 3,$$
$$x_{obsv1} = x - r_s,\ y_{obsv1} = y + 4.$$

Для случая движения в сектор II ($-\varphi_2 \leq \varphi < -\varphi_1$):

$$x_{obsv1} = x - r_s,\ y_{obsv1} = y + r_s, x_{obsv1} = x - r_s + 1,\ y_{obsv1} = y + r_s + 1, \quad (2.35)$$
$$x_{obsv1} = x - r_s + 2,\ y_{obsv1} = y + r_s + 2, x_{obsv1} = x - r_s + 3,\ y_{obsv1} = y + r_s + 3,$$
$$x_{obsv1} = x - r_s - 1,\ y_{obsv1} = y + r_s - 1, x_{obsv1} = x - r_s - 2,\ y_{obsv1} = y + r_s - 2,$$
$$x_{obsv1} = x - r_s - 3,\ y_{obsv1} = y + r_s - 3, x_{obsv1} = x - r_s - 3,\ y_{obsv1} = y + r_s - 3,$$
$$x_{obsv1} = x - r_s - 3,\ y_{obsv1} = y + r_s - 3.$$

Для случая движения в сектор VIII ($\varphi_1 < \varphi \leq \varphi_2$):

$$x_{obsv1} = x - r_s,\ y_{obsv1} = y - r_s, x_{obsv1} = x - r_s + 1,\ y_{obsv1} = y - r_s - 1, \quad (2.36)$$
$$x_{obsv1} = x - r_s + 2,\ y_{obsv1} = y - r_s - 2, x_{obsv1} = x - r_s + 3,\ y_{obsv1} = y - r_s - 3,$$
$$x_{obsv1} = x - r_s - 1,\ y_{obsv1} = y - r_s + 1, x_{obsv1} = x - r_s - 2,\ y_{obsv1} = y - r_s + 2,$$
$$x_{obsv1} = x - r_s - 3,\ y_{obsv1} = y - r_s + 3, x_{obsv1} = x - r_s - 3,\ y_{obsv1} = y - r_s + 3,$$
$$x_{obsv1} = x - r_s - 3,\ y_{obsv1} = y - r_s + 3.$$

Для случая движения в сектор III ($-\varphi_3 \leq \varphi < -\varphi_2$):

$$x_{obsv1} = x,\ y_{obsv1} = y + r_s, x_{obsv1} = x - 1,\ y_{obsv1} = y + r_s, \quad (2.37)$$
$$x_{obsv1} = x - 2,\ y_{obsv1} = y + r_s, x_{obsv1} = x - 3,\ y_{obsv1} = y + r_s,$$
$$x_{obsv1} = x - 4,\ y_{obsv1} = y + r_s, x_{obsv1} = x + 1,\ y_{obsv1} = y + r_s,$$
$$x_{obsv1} = x + 2,\ y_{obsv1} = y + r_s, x_{obsv1} = x + 3,\ y_{obsv1} = y + r_s,$$
$$x_{obsv1} = x + 4,\ y_{obsv1} = y + r_s.$$

Для случая движения в сектор VII ($\varphi_2 < \varphi \leq \varphi_3$):

$$x_{obsv1} = x,\ y_{obsv1} = y - r_s, x_{obsv1} = x - 1,\ y_{obsv1} = y - r_s, \quad (2.38)$$
$$x_{obsv1} = x - 2,\ y_{obsv1} = y - r_s, x_{obsv1} = x - 3,\ y_{obsv1} = y - r_s,$$
$$x_{obsv1} = x - 4,\ y_{obsv1} = y - r_s, x_{obsv1} = x + 1,\ y_{obsv1} = y - r_s,$$
$$x_{obsv1} = x + 2,\ y_{obsv1} = y - r_s, x_{obsv1} = x + 3,\ y_{obsv1} = y - r_s,$$

$$x_{obsv1} = x + 4,\ y_{obsv1} = y - r_s.$$

Для случая движения в сектор IV ($-\varphi_4 \le \varphi < -\varphi_3$):

$$x_{obsv1} = x + r_s - 3,\ y_{obsv1} = y + r_s + 3, x_{obsv1} = x + r_s - 2,\ y_{obsv1} = y + r_s + 2, \quad (2.39)$$

$$x_{obsv1} = x + r_s - 1,\ y_{obsv1} = y + r_s + 1, x_{obsv1} = x + r_s,\ y_{obsv1} = y + r_s,$$

$$x_{obsv1} = x + r_s + 1,\ y_{obsv1} = y + r_s - 1, x_{obsv1} = x + r_s + 2,\ y_{obsv1} = y + r_s - 2,$$

$$x_{obsv1} = x + r_s + 3,\ y_{obsv1} = y + r_s - 3, x_{obsv1} = x + r_s + 3,\ y_{obsv1} = y + r_s - 3,$$

$$x_{obsv1} = x + r_s + 3,\ y_{obsv1} = y + r_s - 3.$$

Для случая движения в сектор VI ($\varphi_3 < \varphi \le \varphi_4$):

$$x_{obsv1} = x + r_s - 3,\ y_{obsv1} = y - r_s - 3, x_{obsv1} = x + r_s - 2,\ y_{obsv1} = y - r_s - 2, \quad (2.40)$$

$$x_{obsv1} = x + r_s - 1,\ y_{obsv1} = y - r_s - 1, x_{obsv1} = x + r_s,\ y_{obsv1} = y - r_s,$$

$$x_{obsv1} = x + r_s + 1,\ y_{obsv1} = y - r_s + 1, x_{obsv1} = x + r_s + 2,\ y_{obsv1} = y - r_s + 2,$$

$$x_{obsv1} = x + r_s + 3,\ y_{obsv1} = y - r_s + 3, x_{obsv1} = x + r_s + 3,\ y_{obsv1} = y - r_s + 3,$$

$$x_{obsv1} = x + r_s + 3,\ y_{obsv1} = y - r_s + 3.$$

Для случая движения в сектор V ($|\varphi| > \varphi_4$):

$$x_{obsv1} = x + r_s,\ y_{obsv1} = y, x_{obsv1} = x + r_s,\ y_{obsv1} = y - 1, \quad (2.41)$$

$$x_{obsv1} = x + r_s,\ y_{obsv1} = y - 2, x_{obsv1} = x + r_s,\ y_{obsv1} = y - 3,$$

$$x_{obsv1} = x + r_s,\ y_{obsv1} = y - 4, x_{obsv1} = x + r_s,\ y_{obsv1} = y + 1,$$

$$x_{obsv1} = x + r_s,\ y_{obsv1} = y + 2, x_{obsv1} = x + r_s,\ y_{obsv1} = y + 3,$$

$$x_{obsv1} = x + r_s,\ y_{obsv1} = y - r_s + 4.$$

Проводя аналогичные рассуждения для случая $V>V_3$, когда виртуальное препятствие расположено под острым углом к роботу, получаем следующие выражения.

Для случая движения в сектор I ($|\varphi| \le \varphi_1$):

$$x_{obsv1} = x - r_s - 1,\ y_{obsv1} = y, x_{obsv1} = x - r_s,\ y_{obsv1} = y - 1, \quad (2.42)$$

$$x_{obsv1} = x - r_s + 1,\ y_{obsv1} = y - 2, x_{obsv1} = x - r_s + 2,\ y_{obsv1} = y - 3,$$

$$x_{obsv1} = x - r_s + 3,\ y_{obsv1} = y - 4, x_{obsv1} = x - r_s,\ y_{obsv1} = y + 1,$$

$$x_{obsv1} = x - r_s + 1,\ y_{obsv1} = y + 2, x_{obsv1} = x - r_s + 2,\ y_{obsv1} = y + 3,$$

$$x_{obsv1} = x - r_s + 3,\ y_{obsv1} = y + 4.$$

Для случая движения в сектор II ($-\varphi_2 \le \varphi < -\varphi_1$):

$$x_{obsv1} = x - r_s - 1,\ y_{obsv1} = y + r_s + 1, x_{obsv1} = x - r_s,\ y_{obsv1} = y + r_s + 1, \quad (2.43)$$

$$x_{obsv1} = x - r_s + 1,\ y_{obsv1} = y + r_s + 1, x_{obsv1} = x - r_s + 2,\ y_{obsv1} = y + r_s + 1,$$

$$x_{obsv1} = x - r_s + 3,\ y_{obsv1} = y + r_s + 1, x_{obsv1} = x - r_s - 1,\ y_{obsv1} = y + r_s,$$

$$x_{obsv1} = x - r_s - 1,\ y_{obsv1} = y + r_s - 1, x_{obsv1} = x - r_s - 1,\ y_{obsv1} = y + r_s - 2,$$

$$x_{obsv1} = x - r_s - 1,\ y_{obsv1} = y + r_s - 3.$$

Для случая движения в сектор VIII ($\varphi_1 < \varphi \leq \varphi_2$):

$$x_{obsv1} = x - r_s - 1,\ y_{obsv1} = y - r_s - 1, x_{obsv1} = x - r_s,\ y_{obsv1} = y - r_s - 1, \quad (2.44)$$

$$x_{obsv1} = x - r_s + 1,\ y_{obsv1} = y - r_s - 1, x_{obsv1} = x - r_s + 2,\ y_{obsv1} = y - r_s - 1,$$

$$x_{obsv1} = x - r_s + 3,\ y_{obsv1} = y - r_s - 1, x_{obsv1} = x - r_s - 1,\ y_{obsv1} = y - r_s,$$

$$x_{obsv1} = x - r_s - 1,\ y_{obsv1} = y - r_s + 1, x_{obsv1} = x - r_s - 1,\ y_{obsv1} = y - r_s + 2,$$

$$x_{obsv1} = x - r_s - 1,\ y_{obsv1} = y - r_s + 3.$$

Для случая движения в сектор III ($-\varphi_3 \leq \varphi < -\varphi_2$):

$$x_{obsv1} = x,\ y_{obsv1} = y + r_s + 1, x_{obsv1} = x - 1,\ y_{obsv1} = y + r_s, \quad (2.45)$$

$$x_{obsv1} = x - 2,\ y_{obsv1} = y + r_s - 1, x_{obsv1} = x - 3,\ y_{obsv1} = y + r_s - 2,$$

$$x_{obsv1} = x - 4,\ y_{obsv1} = y + r_s - 3, x_{obsv1} = x + 1,\ y_{obsv1} = y + r_s,$$

$$x_{obsv1} = x + 2,\ y_{obsv1} = y + r_s - 1, x_{obsv1} = x + 3,\ y_{obsv1} = y + r_s - 2,$$

$$x_{obsv1} = x + 4,\ y_{obsv1} = y + r_s - 3.$$

Для случая движения в сектор VII ($\varphi_2 < \varphi \leq \varphi_3$):

$$x_{obsv1} = x,\ y_{obsv1} = y - r_s - 1, x_{obsv1} = x - 1,\ y_{obsv1} = y - r_s, \quad (2.46)$$

$$x_{obsv1} = x - 2,\ y_{obsv1} = y - r_s + 1, x_{obsv1} = x - 3,\ y_{obsv1} = y - r_s + 2,$$

$$x_{obsv1} = x - 4,\ y_{obsv1} = y - r_s + 3, x_{obsv1} = x + 1,\ y_{obsv1} = y - r_s,$$

$$x_{obsv1} = x + 2,\ y_{obsv1} = y - r_s + 1, x_{obsv1} = x + 3,\ y_{obsv1} = y - r_s + 2,$$

$$x_{obsv1} = x + 4,\ y_{obsv1} = y - r_s + 3.$$

Для случая движения в сектор IV ($-\varphi_4 \leq \varphi < -\varphi_3$):

$$x_{obsv1} = x + r_s + 1,\ y_{obsv1} = y + r_s + 1, x_{obsv1} = x + r_s,\ y_{obsv1} = y + r_s + 1, \quad (2.47)$$

$$x_{obsv1} = x + r_s - 1,\ y_{obsv1} = y + r_s + 1, x_{obsv1} = x + r_s - 1,\ y_{obsv1} = y + r_s + 1,$$

$$x_{obsv1} = x + r_s - 3,\ y_{obsv1} = y + r_s + 1, x_{obsv1} = x + r_s + 1,\ y_{obsv1} = y + r_s,$$

$$x_{obsv1} = x + r_s + 1,\ y_{obsv1} = y + r_s - 1, x_{obsv1} = x + r_s + 1,\ y_{obsv1} = y + r_s - 2,$$

$$x_{obsv1} = x + r_s + 1,\ y_{obsv1} = y + r_s - 3.$$

Для случая движения в сектор VI ($\varphi_3 < \varphi \leq \varphi_4$):

$$x_{obsv1} = x + r_s + 1,\ y_{obsv1} = y - r_s - 1, x_{obsv1} = x + r_s,\ y_{obsv1} = y - r_s - 1, \quad (2.48)$$

$$x_{obsv1} = x + r_s - 1,\ y_{obsv1} = y - r_s - 1, x_{obsv1} = x + r_s - 2,\ y_{obsv1} = y - r_s - 1,$$

$$x_{obsv1} = x + r_s - 3,\ y_{obsv1} = y - r_s - 1, x_{obsv1} = x + r_s + 1,\ y_{obsv1} = y - r_s,$$

$$x_{obsv1} = x + r_s + 1,\ y_{obsv1} = y - r_s + 1, x_{obsv1} = x + r_s + 1,\ y_{obsv1} = y - r_s + 2,$$

$$x_{obsv1} = x + r_s + 1,\ y_{obsv1} = y - r_s + 3.$$

Для случая движения в сектор V ($|\varphi| > \varphi_4$):

$$x_{obsv1} = x + r_s + 1,\ y_{obsv1} = y, x_{obsv1} = x + r_s,\ y_{obsv1} = y - 1, \quad (2.49)$$

$$x_{obsv1} = x + r_s - 1,\ y_{obsv1} = y - 2, x_{obsv1} = x + r_s - 2,\ y_{obsv1} = y - 3,$$

$$x_{obsv1} = x + r_s - 3,\ y_{obsv1} = y - 4, x_{obsv1} = x + r_s,\ y_{obsv1} = y + 1,$$

$$x_{obsv1} = x + r_s - 1,\ y_{obsv1} = y + 2, x_{obsv1} = x + r_s - 2,\ y_{obsv1} = y + 3,$$

$$x_{obsv1} = x + r_s - 3,\ y_{obsv1} = y + 4.$$

После введения виртуальных препятствий осуществляется планирование пути методом быстро растущих случайных деревьев RRT. Процедура поиска пути с использованием алгоритма RRT состоит в циклическом обновлении карты среды, выполнении функции распространения случайно растущего дерева и поиске пути на полученном случайном графе.

В данной диссертации разработана модифицированная версия алгоритма RRT, отличающаяся от оригинала тем, что в нем используется несколько родительских узлов. Данная версия алгоритма описывается следующим псевдокодом.

Алгоритм 1 построения случайного графа для планирования траектории движения мобильного робота.

Входы: Положение робота (x, y); карта среды Map; целевая точка (x_t, y_t); скорость робота V; угол рысканья φ; параметр r_s; параметр оптимизации N_{add}.

Выходы: Список узлов дерева Tree; матрица связей Matrix_tr.

1: [Tree, Matrix_tr] ← Init_Tree([x,y], [xt,yt]);
2: Map←Save_Vir_Obs(V, φ);
3: FL_0←Str_Line(Tree(last),[xt, yt], Map, Ls);
4: **IF** FL_0=1
5: **WHILE** FL_1<Nadd

```
6:    X_rand←Rand_Node(Map, Tree, Ls)
7:    X_nod1←Parent_Node(Tree, Map,rs, []);
8:    X_nod2←Parent_Node(Tree, Map, rs, X_nod1);
9:    Tree←Expand_Tree(X_rand);
10:   Matrix_tr← Expand_Matr(X_nod1, X_rand);
11:   Matrix_tr← Expand_Matr(X_nod2, X_rand);
12:   FL_2←Str_Line(X_rand, [xt,yt], Map, Ls);
13:    IF FL_2=1
14:      FL_1= FL_1+1;
15:      Matrix_tr← Expand_Matr(X_rand,[ xt,yt]);
16:    END IF
17:  END WHILE
18. ELSE
19:   Matrix_tr← Expand_Matr([x,y],[xt,yt]);
20: END IF
21: Path_opt=Aco(Matrix_tr, Tree);
```

Функция Init_Tree () инициализирует дерево и матрицу смежности Matrix_tr. В процессе фазы инициализации осуществляется добавление начальной точки (x, y) и целевой точки (x_t, y_t) в список узлов дерева. Первый узел дерева – всегда начальное положение робота (x, y). Последним узлом дерева является целевая точка (x_t, y_t). Матрица связей Matrix_tr инициализируется как нулевая матрица размерности 2×2.

Функция Save_Vir_Obs добавляет виртуальные препятствия на карту в соответствии с выражениями (2.18) – (2.49) и текущей линейной скоростью и направлением движения робота. При этом карта среды представляет собой представленную на рис. 2.16 матрицу размерности $N_x \times N_y$, в которой число ячеек определяется выражением

$$Nx = \lceil Lx/Sx \rceil,\ Ny = \lceil Ly/Sy \rceil, \qquad (2.50)$$

где S_x, S_y – размеры ячеек пространства по осям OX_g и OY_g, соответственно; $\lceil * \rceil$ – операция округления до ближайшего сверху целого числа.

В строке 3 Алгоритма 1 проверяется возможность построить отрезок, соединяющий точки (x, y) и (x_t, y_t) без коллизий. Такая проверка проводится с помощью функции Str_Line(), которая возвращает флаг FL_0, равный 1,

если присутствует коллизия. Коллизией считается приближение траектории движения робота к любому из препятствий ближе, чем на расстояние L_s.

Основной цикл построения случайного дерева начинается с генерирования случайного узла X_rand, для чего используется функция Rand_Node(). Указанная функция генерирует узел таким образом, чтобы расстояние от него до препятствий и других узлов дерева составляло не менее L_s. Далее осуществляется поиск нового родительского узла с помощью функции Parent_Node(). Использование двух родительских узлов является ключевым отличием данного алгоритма от традиционной версии RRT. Основной планируемый эффект от введения двух родительских узлов – снижение длины планируемого с помощью случайного пути. Вместе с тем, можно ожидать некоторое увеличение времени построения дерева. Снижение длины планируемого пути обусловлено тем, что увеличение числа родительских узлов увеличивает число связей в строящемся графе, что делает возможным более эффективную оптимизацию при поиске кратчайшего пути.

Если родительские узлы найдены успешно, то новый узел X_rand добавляется в дерево Tree, а в матрицу Matrix_tr добавляются соответствующие связи, для чего используются функции Expand_Tree() и Expand_Matr() соответственно.

Далее, используя функцию Str_Line() производится проверка наличия коллизий на отрезке, соединяющем новый узел X_rand и целевую точку (x_t, y_t). Если путь без коллизий найден, то происходит инкремент счетчика FL_1. Цикл построения дерева завершается после того, как будет найдено N_{add} узлов, имеющих прямой путь в целевую точку.

Поиск кратчайшего пути на построенном графе может осуществляться либо алгоритмом Дейкстры [184], [185, с. 177-183], либо с использованием эвристических процедур. Численные исследования показали, что

для достаточно сложной среды (с плотностью препятствий свыше 20 %) более эффективны численные процедуры. В этой связи поиск кратчайшего пути осуществляется с помощью муравьиного алгоритма [186]. Выбор эволюционного алгоритма обусловлен необходимостью оптимизировать время поиска кратчайшего пути. Поиск кратчайшего пути (строка 21 Алгоритма 1), базирующийся на применении муравьиного алгоритма, реализуется с помощью функции Aco(). Входом данной функции является дерево Tree и матрица смежности Matrix_tr. Выходом функции Aco () является список узлов, составляющих оптимальный путь Path_opt. Этот список определяет кривую, состоящую из прямолинейных отрезков. Данную кривую можно сгладить и оптимизировать локальными методами.

После определения целевой точки в текущий момент времени осуществляется вычисление требуемого направления движения и круговой частоты вращения робота в соответствии с выражениями [200]

$$\varphi_r = actg\frac{y_t - y}{x_t - x},\ \omega_r = k_\omega(\varphi_r - \varphi), \qquad (2.51)$$

где (x_t, y_t) – координаты целевой точки; (x, y) – координаты робота.

2.3.3 Оптимизация и сглаживание траектории движения

В силу достаточно медленной сходимости траектории, найденной методов RRT, к оптимальному решению, примем двухэтапный метод планирования движения робота. На первом этапе, используя метод RRT находится субоптимальная траектория первого приближения. Дальнейшая оптимизация проводится методами локальной оптимизации, которые существенно быстрее методов глобальной оптимизации. В данной работе используется подход, предложенный В.А. Костюковым в работах [27, 187]. Этот подход позволяет учесть не только препятствия, но и конфликтные области, которые находятся в среде функционирования робота.

Согласно работам [27, 187] оптимизационная процедура базируется на функционале, включающем в себя две составляющие

$$G(Tr) = \delta_1 \sum_{i=2}^{N-1} \left[\left(x_i - x_i^{'}\right)^2 + \left(y_i - y_i^{'}\right)^2 \right] + \delta_2 \sum_{i=2}^{N-1} [(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2], \quad (2.52)$$

где первое слагаемое – сумма квадратов отклонений точек (x_i, y_i) искомой траектории от точек траектории первого приближения (x'_i, y'_i); второе слагаемое – длина оптимизируемой траектории T_r; коэффициенты $0 \le \delta_1, \delta_2 \le 1$ – веса каждого слагаемого.

Стационарная точка функционала (2.52) по переменным $\{x_i, y_i\}$, i = 2, 3, ..., N-1, находится из следующей системы уравнений

$$\begin{cases} \partial G(Tr)/\partial x_i = 0, \\ \partial \frac{G(Tr)}{\partial} y = 0, \ i = 2, \dots, N-2 \end{cases} \quad (2.53)$$

Система (2.53) распадается на две независимые подсистемы линейных уравнений относительно векторов $X = [x_2 \dots x_{N-1}]^T$ и $Y = [y_2 \dots y_{N-1}]^T$. Решения этих двух подсистем представляются в виде

$$\begin{aligned} X_{zt} &= [x_{st2} \dots x_{stN-1}]^T = A^{-1} B_X, \\ Y_{zt} &= [y_{st2} \dots y_{stN-1}]^T = A^{-1} B_Y, \end{aligned} \quad (2.54)$$

$$A = \begin{bmatrix} \delta_1 + 2\delta_2 & -\delta_2 & 0 & 0 & \dots & 0 & 0 & 0 \\ -\delta_2 & \delta_1 + 2\delta_2 & -\delta_2 & 0 & \dots & . & 0 & 0 \\ 0 & -\delta_2 & \delta_1 + 2\delta_2 & -\delta_2 & \dots & 0 & . & 0 \\ 0 & 0 & -\delta_2 & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & . & . & 0 & 0 \\ 0 & . & 0 & \dots & . & \delta_1 + 2\delta_2 & -\delta_2 & 0 \\ 0 & 0 & . & \dots & 0 & -\delta_2 & \delta_1 + 2\delta_2 & -\delta_2 \\ 0 & 0 & 0 & \dots & 0 & 0 & -\delta_2 & \delta_1 + 2\delta_2 \end{bmatrix}, \quad (2.55)$$

$$B_x = \left[\delta_1 x_2^{'} + \delta_2 x_1^{'}, \delta_1 x_3^{'}, \dots, \delta_1 x_{N-2}^{'}, \delta_1 x_{N-1}^{'} + \delta_2 x_N^{'}\right]^T \quad (2.56)$$

$$B_x = \left[\delta_1 y_2^{'} + \delta_2 y_1^{'}, \delta_1 y_3^{'}, \dots, \delta_1 y_{N-2}^{'}, \delta_1 y_{N-1}^{'} + \delta_2 y_N^{'}\right]^T \quad (2.57)$$

Решение (2.54) удовлетворяет достаточным условиям локального максимума по отношению к минорам матрицы Гессе при таких значениях коэффициентов δ_1, δ_2, которые подчинены неравенствам $0 \le \delta_1, \delta_2 \le 1$, $\delta_G + \delta_L = 1$. Действительно, легко показать, что матрица Гессе для функционала (2.52) является блочно-диагональной матрицей вида

$$Gess(G) = \begin{bmatrix} A & 0_{N-2 \times N-2} \\ 0_{N-2 \times N-2} & A \end{bmatrix}, \quad (2.58)$$

где $0_{N-2 \times N-2}$ – нулевая матрица размерности N-2×N-2.

Известно, что, если каждый блок блочно-диагональной матрицы является положительно определенной матрицей, то она также является положительно определенной [183]. Также, согласно следствию из теориме Гершгорина [183], для любой симметрической матрицы с положительными элементами на главной диагонали достаточные условия положительной определенности этой матрицы заключаются в строгом доминировании диагональных элементов. Т.к. матрица (2.55) является симметрической тридиагональной матрицей Теплица, то условие строгого доминирования диагональных элементов принимает вид

$$|\delta_1 + 2\delta_2| > 2|\delta_2|, \tag{2.58}$$

Т.к. $0 \le \delta_1, \delta_2 \le 1$, то из выражения (2.58) следует

$$\delta_1 + 2\delta_2 > 2\delta_2 \Rightarrow \delta_1 > 0. \tag{2.59}$$

Отметим, что в точке (2.54) достигается и глобальный минимум функционала (2.52), т.к. функционал квадратичный, т.е. имеет единственный экстремум.

Для минимизации длины результирующей кривой используются два дополнительных условия. Первое условие состоит в отсутствии коллизий с препятствиями. Второе условие заключается в ограничении максимального отклонения от исходной кривой первого приближения, которое может быть записано в виде

$$\sqrt{\sum_{i=2}^{N-1}\left[\left(x_i - x_i^{'}\right)^2 + \left(y_i - y_i^{'}\right)^2\right]/(N-2)} \le \sigma_{kr}. \tag{2.60}$$

С учетом вышесказанного процедура нахождения пути минимальной длины имеет, учитывающая указанные ограничения, имеет следующий вид. Во-первых, рассматриваются последовательно значения коэффициента δ_1 из диапазона $[\delta_{1min}, \delta_{1max}]$ с шагом $\Delta\delta_1 << \delta_{1max}$. Значение $\delta_1=\delta_{1min}$ соответствует случаю, когда оптимизируемая траектория наиболее близка, в смысле среднеквадратичного отклонения, к прямой линии, соединяющей начальную и конечную точки пути. Если коллизии отсутствуют и условие

(2.60) не нарушается, то траектория при $\delta_1=\delta_{1min}$ является наилучшей. В противном случае значение δ_1 увеличивается.

Это процедура повторяется итеративно, пока не будет найдено значение δ_1, для которого перечисленные условия будут выполнены. Чтобы поиск завершался успешно необходимо выбирать шаг $\Delta\delta_1$ таким образом, чтобы он соответствовал верхней границе σ_{kr}.

После нахождения оптимизированного пути, он подвергается сглаживанию. Для этого применяется сопряжение отрезков, соединяющих отдельные точки траектории, дугами окружностей. Такие дуги окружностей выбираются для всех узловых точек траектории.

Результирующая траектория движения в виде списка точек, которые должен пройти робот, поступает в систему управления, которая вычисляет желаемое направление и скорость движения.

2.4 Исследование алгоритмов планирования численными методами

Принята среда размерами Lx=100 м и Ly=100 м. В данной среде генерируется 24 прямоугольных препятствия. Положение левого нижнего угла каждого препятствия генерируется случайным образом. Размеры препятствий также генерируются случайным образом с помощью равномерного закона распределения. При этом ширина и длина могут составлять от 1 до 11 метров каждая сторона. Одиночные препятствия могут конфигурироваться в более сложные формы. Среда может быть интерпретирована как городская среда плотностью застройки до 30 %.

Параметры уравнений динамики подвижного объекта: a_v=3; a_w=5. Приняты ограничения на максимальную линейная и угловую скорости подвижного объекта, равные V_{max} = 4 м/с, ω_{max} = 2 рад/с. Расстояние от подвижного объекта до препятствия, определяющее момент, в который робот начинает сбрасывать свою скорость, равно r_s = 6 м. Начальные положение

и ориентация мобильного робота в среде задаются случайным образом. При этом начальное положение робота расположено на удалении не менее 4 м от препятствий.

Целевая точка также генерируется случайным образом. При этом минимальное расстояние от целевой точки до препятствий также составляет не менее 6 м. При этом требований к ориентации робота в целевой точке не предъявляется.

Найденный путь описывается списком узлов, которые должен пройти робот, а также списком пройденных узлов. Т.к. первая точка совпадает с начальным положением робота, то она сразу заносится в список пройденных точек и движение осуществляется во вторую точку. Далее в цикле осуществляется определение расстояния до узлов, находящихся в списке запланированного пути. Данный список представляет собой вектор, содержащий 0, если узел еще не пройден и 1, если узел пройден. По этому списку и расстоянию до него и следующего узла определяется номер последнего пройденного узла и номер узла, в который в настоящий момент должен двигаться робот.

2.4.1 Исследование влияния числа узлов на длину и время расчета траектории

Исследование влияния числа узлов на длину и время расчета траектории проведено для числа родительских узлов 1, 2, 3 и значений параметра N_{add} = 20, 40, 80. Для каждого числа узлов и значения N_{add} собрана статистика из 1000 экспериментов, результаты которых сведены в Табл. 2.1 и рис. 2.20 и 2.21.

Параметр N_{add} определяет степень оптимальности траектории, планируемой с применением алгоритма случайных деревьев RRT. Чем больше этот параметр, тем выше вероятность того, что полученная траектория приближается к оптимальной.

Усредненная относительная длина пути рассчитывается в соответствии с выражением

$$Dl = \frac{1}{1000}\sum_{i=1}^{1000}(Dl_i/r_{rti}). \quad (2.61)$$

где Dl_i – длина l-го пути в i-м эксперименте; r_{rti} – расстояние между начальным положением робота и целевой точкой в i-м эксперименте.

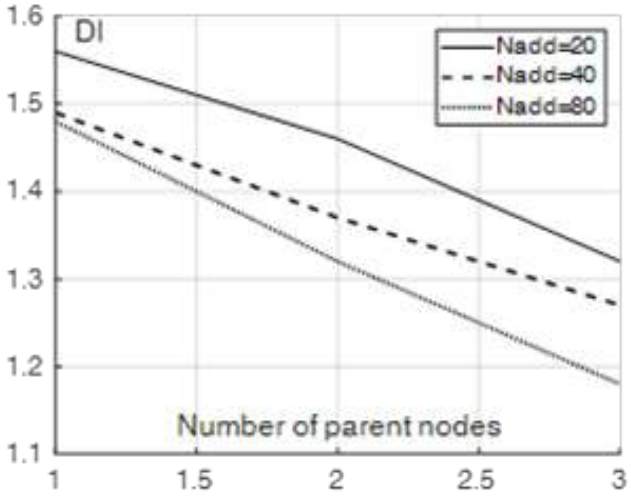


Рисунок 2.20 – Зависимость усредненной относительной длины пути от количества родительских узлов

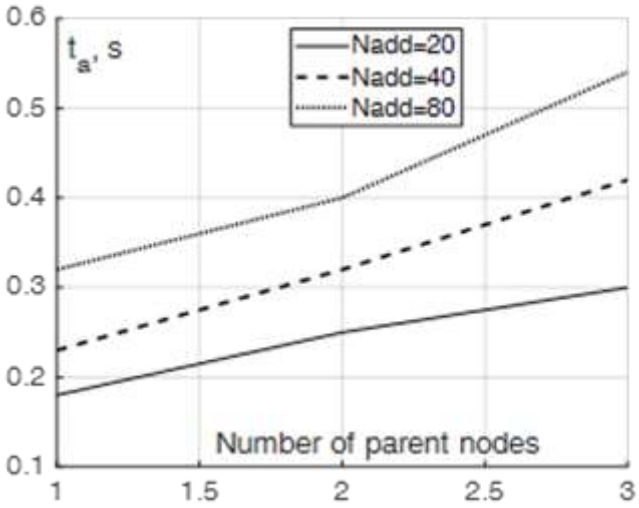


Рисунок 2.21 – Усредненное время расчета траектории подвижного объекта от количества родительских узлов

Время расчета траектории в каждом эксперименте определено с помощью функций tic и toc, встроенных в Matlab. В таблице 2.1 представлено

усредненное по 1000 экспериментов среднее время, вычисляемое в соответствии с выражением

$$D t_a = \frac{1}{1000} \sum_{i=1}^{1000} t_i, \quad (2.62)$$

где t_i – iвремя расчета траектории в i-м эксперименте.

Таблица 2.1 – Влияние числа родительских узлов на длину планируемого пути и время расчета траектории

Число узлов	Параметр N_{add}	Средняя длина пути	Среднее время расчета
1 узел	80	1.48	0.32
	40	1.49	0.23
	20	1.56	0.18
2 узла	80	1.32	0.4
	40	1.37	0.32
	20	1.46	0.25
3 узла	80	1.18	0.54
	40	1.27	0.42
	20	1.32	0.3

Из табл. 2.1 можно сделать вывод, что длина планируемой траектории уменьшается с ростом количества родительских узлов, используемых в алгоритме RRT. Это связано с тем, что число родительских узлов определяет число связей каждого узла случайного графа. Соответственно, число возможных путей из текущего положения робота (x, y) в целевую точку (x_t, y_t) также растет.

Для одинаковых значений параметра N_{add} длина планируемого пути, по сравнению с оригинальным алгоритмом RRT, снижается на 6.5 – 11 % для варианта с двумя родительскими узлами, и на 15 – 20 % для варианта с тремя родительскими узлами.

Аналогичный анализ для одинаковых значений параметра N_{add} показывает, что уменьшение длины пути сопровождается увеличением времени расчета траектории на 28 – 40 процентов для варианта с двумя родительскими узлами, и на 40 – 45 % для варианта с тремя родительскими узлами.

Вместе с тем, параметр N_{add} является внутренним оптимизационным параметром алгоритма RRT. Если сравнивать время расчета пути одинаковой длины, то можно получить следующие результаты. Из табл. 2.1 следует, что длина пути оригинального алгоритма RRT с одним родительским узлом, обеспечиваемая при N_{add} =80, соответствует длине пути алгоритма RRT с двумя родительскими узлами, при N_{add} =20. При этом время расчета траектории модифицированным алгоритмом RRT примерно на 20 % меньше, чем оригинальным алгоритмом RRT. Аналогичным образом, сравнивая время расчета алгоритмов RRT с двумя и тремя родительскими узлами при средней относительной длине пути 1,32 получаем, что алгоритм в тремя узлами находит путь той же длины, то и алгоритм с двумя узлами на 25 % быстрее.

2.4.2 Исследование влияния параметров локальной оптимизации на планируемую траекторию

К параметрам метода локальной оптимизации и сглаживания относятся коэффициенты функционала (2.60) и параметр σ_{kr}. При выполнении условия $\delta_1+\delta_2=1$, в функционале (2.60) остается один независимый коэффициент, который выбирается в соответствии с итерационной процедурой, описанной в разделе 2.3.3. В этой связи ключевым параметром, существенно влияющим на вид оптимизированной траектории, является допустимое среднеквадратичное отклонение σ_{kr}.

Результаты моделирования алгоритма локальной оптимизации и сглаживания траектории при σ_{kr} = 10 и σ_{kr} = 5, представлены на рис. 2.22 и рис. 2.23 соответственно.

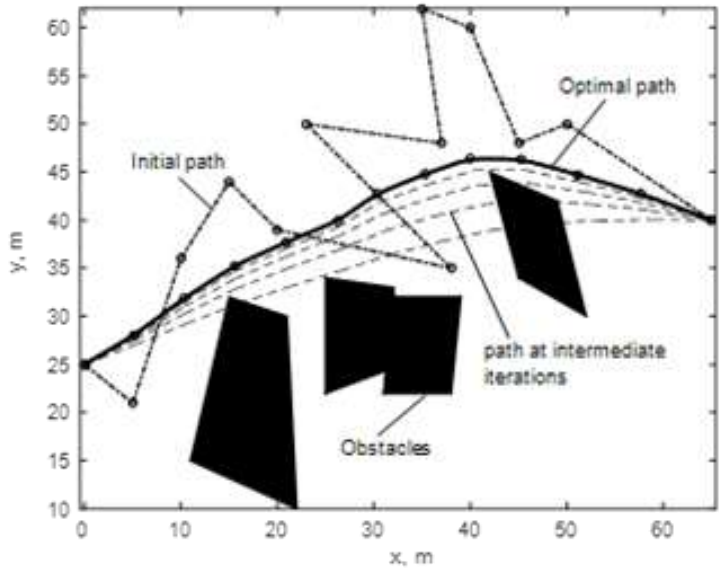


Рисунок 2.22 – Оптимизация траектории при σ_{kr} = 10.

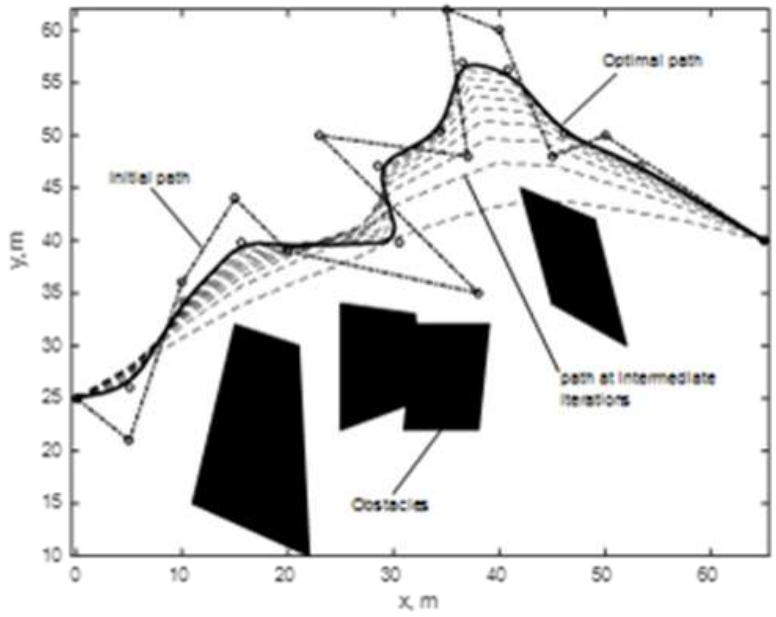


Рисунок 2.23 – Оптимизация траектории при σ_{kr} = 5.

Длина исходной траектории равна 148.5 м. В результате применения процедуры сглаживания и оптимизации получена траектория длиной 71.6 м при σ_{kr} = 10 и длиной 90.8 м при σ_{kr} = 5. При этом параметр δ_1 изменялся в интервале [0.02 0.98] с шагом 0.02.

На основании проведенных исследований можно сделать вывод о том, что с ужесточением ограничений на отклонение от исходной кривой , т.е. с уменьшением значения σ_{kr}, длина траектории увеличивается.

2.4.3 Моделирование движения робота в среде с препятствиями

ТРезультаты моделирования системы управления движением мобильного робота с описанными в данной главе алгоритмами планирования представлены на рис. 2.27 – 2.27. На рис. 2.24 представлена среда, содержащая реальные препятствия и виртуальное препятствие в начальный момент времени. Положение робота в начальный момент времени обозначено символом "*". Целевое положение робота обозначено символом "+". Путь, найденный с помощью алгоритма RRT с двумя родительскими узлами обозначен пунктирной линией, а оптимизированная траектория показана сплошной линией. Для планирования RRT алгоритм сгенерировал граф, включающий 341 узел. В результате поисковой процедуры, основанной на муравьином алгоритме, найден путь, содержащий 8 узлов, включая стартовую и целевую точки:

$$Tr = \begin{bmatrix} 18 & 26 & 33 & 41 & 50 & 50 & 48 & 93 \\ 84 & 90 & 96 & 91 & 98 & 94 & 93 & 52 \end{bmatrix}. \quad (2.63)$$

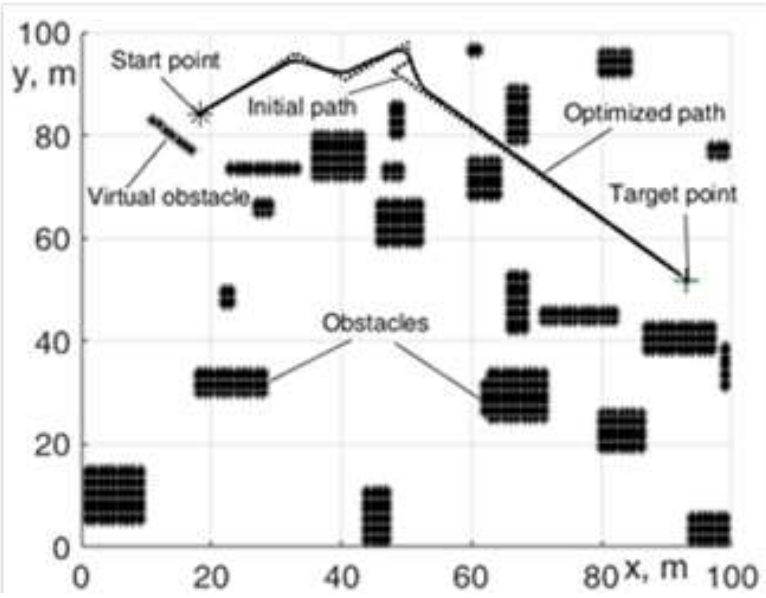
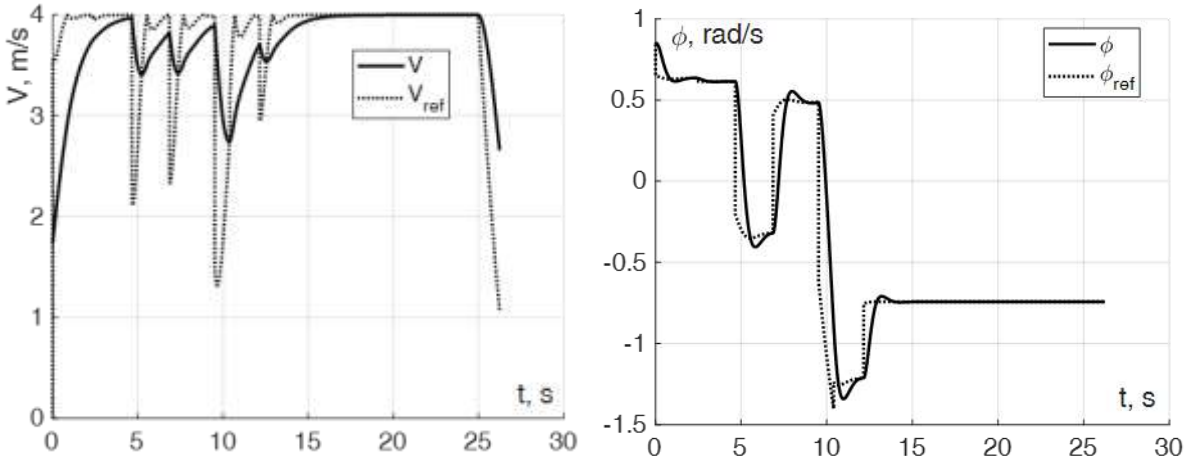


Рисунок 2.24 – Траектория движения

Длина пути, полученная с помощью алгоритма RRT составила примерно 107 м. Длина оптимизированного пути составила 99.9 м. Таким образом, процедура локальной оптимизации позволила сократить длину пути на 6.6%. Исследования показали, что с ростом числа промежуточных точек

между начальным и целевым положением робота эффект применения локальной процедуры оптимизации увеличивается. Для траекторий, состоящих из двух и трех точек, процедура локальной оптимизации не эффективна.

На рис. 2.25 а) показана зависимость скорости робота от времени, а на рис. 2.25 б) – изменение угла рысканья во времени. На интервале времени от 0 до 5 секунд, робот ускоряется до значения V_{max} =4 м/с. Далее на интервале времени от 5 до 13 секунд система управления уменьшает скорость робота 4 раза. Первые три маневра по скорости обусловлены наличием поворотов, а последнее снижение скорости обусловлено прохождением вблизи препятствий. Также скорость движения снижается при приближении к целевой точке. Также, из рис. 2.25 б) следует, что робот в процессе движения осуществляет маневр по курсу в начальный момент времени, а затем трижды, двигаясь между узлами запланированной траектории.



а) Линейная скорость б) Угол рысканья

Рисунок 2.25 – Изменение скорости и угла рысканья мобильного робота

На рис. 2.26 представлен пример движения робота в среде, включающий движение в узком коридоре. Исследование таких ситуаций показало, что присутствие узких коридоров значительно увеличить время планирования траектории робота. В этой связи в диссертации используется отдельный вариант RRT алгоритма, в котором генерирование новых потенциальных

узлов ограничено определенной областью относительно последнего добавленного в граф родительского узла. Такой способ построения случайного графа дает дополнительное уменьшение длины планируемого пути. Так было проведено исследование, в ходе которого сравнены RRT алгоритм с двумя родительскими узлами с областью генерирования новых узлов, совпадающей с областью функционирования, и с областью, ограниченной 20 метрами относительно последнего добавленного узла.

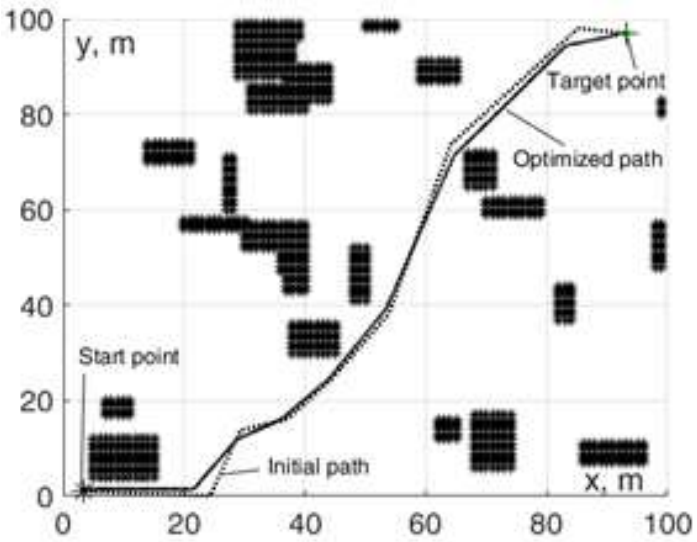


Рисунок 2.26 – Траектория движения мобильного робота в среде с узким коридором

Проведено 1000 экспериментов, в результате которых получено, что ограничение области генерирования новых узлов дает дополнительное уменьшение длины траектории на 5 – 10 %, в зависимости от параметра N_{add}. Однако, данная версия RRT алгоритма может требовать существенно большего времени для расчета траектории, в среднем до 30 % по сравнению с генерированием новых потенциальных узлов во всей области функционирования. Такое увеличение вычислительных затрат наблюдается в среднем. Для ситуаций с узкими проходами алгоритм с локальной генерацией новый потенциальных узлов не приводит к увеличению времени расчета траектории. В этой связи можно рекомендовать использовать блок оценки среды, который определяет необходимость использование того или иного способа

генерирования новых потенциальных узлов пути. Такой блок может быть построен на базе интеллектуальных технологий. Например, в работе [167] представлена нейронная сеть, которая оценивает геометрическую сложность среды.

Отметим, что генерирование нового потенциального узла в ограниченной области приводит к наличию большего числа точек в планируемом пути. При этом, применение описанной выше процедуры локальной оптимизации позволяет уменьшить длину траектории на 10 – 14 % в условиях моделирования. Эффект оптимизации траектории хорошо виден на рис. 2.27, на котором новые потенциальные узлы генерировались в пределах 20 м от последнего добавленного в дерево узла.

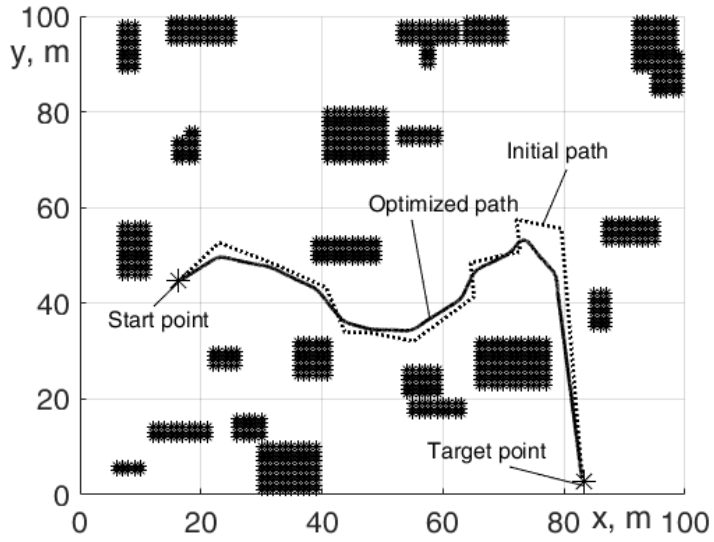


Рисунок 2.27 – Траектория движения мобильного робота при генерировании новых потенциальных узлов в ограниченной области

На рис. 2.28 представлены результаты моделирования движения робота при большой начальной скорости, которые демонстрируют влияние виртуального препятствия на планируемый путь.

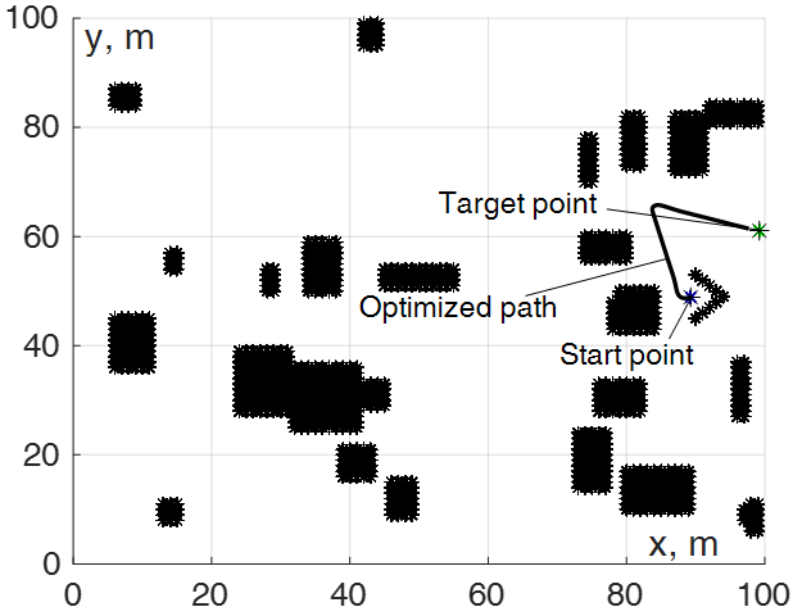


Рисунок 2.28 – Траектория движения мобильного робота с виртуальным препятствием

2.5 Выводы по главе 2

В данной главе представлены следующие новые результаты.

1. Предложен двухэтапный метод планирования траектории движения робота, отличающийся использованием на первом этапе модифицированного алгоритма быстро растущих случайных деревьев для получения траектории первого приближения, а на втором этапе – итерационного алгоритма локальной оптимизации. Применение данной процедуры позволяет снизить вычислительные затраты за счет того, что алгоритм быстро растущих случайных деревьев применятся только для получения грубой траектории первого приближения, что может быть сделано достаточно быстро. Процедура сглаживания и оптимизации траектории осуществляется более быстрыми известными алгоритмами локальной оптимизации.

2. Предложен модифицированный алгоритм быстро растущих случайных деревьев, отличающийся использованием нескольких родительских узлов, что позволяет сократить время расчета траектории на 20 – 25 % по сравнению с оригинальным алгоритмом и при одинаковой степени приближения к оптимальной траектории.

3. Предложен метод учета ограничений на маневры робота, зависящие от его скорости, отличающийся использованием виртуальных препятствий, которые перекрывают физически недоступные для движения сектора, что позволяет учитывать динамические свойства мобильного робота без усложнения алгоритмов планирования, ориентированных на использование карты среды.

Глава 3. Планирование движения с использованием нейросетевых технологий

3.1. Общее описание проблемы планирования движения с использованием нейронных сетей

В данной главе рассматривается метод планирования, базирующийся на использовании нейронной сети. Предлагается использовать нейронную сеть, которая обучается на примерах известных методов планирования пути, но осуществляет не полный расчет траектории движения, а только расчет текущего направления движения робота. Выдвигается гипотеза о том, что совокупность признаков для планирования направления движения, которую кодирует нейронная сеть при обучении, в вычислительном плане окажется менее затратной, чем вычисления в соответствии с известными алгоритмами поиска пути. Это позволит эффективно использовать нейросетевой планировщик в некартографированных или динамических средах, когда маршрут необходимо перестраивать в зависимости от ситуации.

Как известно [189] машинное обучение включает в себя методы контролируемого обучения, неконтролируемое обучение и обучение с подкреплением. При контролируемом обучении в нейронная сеть настраивается таким образом, чтобы наиболее точно повторять алгоритм, использующийся в качестве учителя. В такой постановке нейронная сеть может иметь преимущества перед остальными методами планирования только в вычислительной эффективности. При неконтролируемом обучении и обучении с подкреплением возможно решение задач, которые не решаются другими методами.

При обучении часть выборки от 70 до 90 % предъявляется нейронной сети. Оставшаяся часть выборки не предъявляется нейронной сети и используется при тестировании.

Особое внимание в последнее время уделяется глубокому обучению, которое заключается в применении нейронных сетей с большим число скрытых слоев. Глубокое обучение, предъявляя значительные требования к вычислительным ресурсам, позволяет решать разнообразные задачи в области обработки сигналов и изображений [190, 191], распознавания объектов [192], распознавание речи [193].

Значительное снижение вычислительных затрат при обучении и функционировании нейронных сетей возможно при использовании сверточных сетей [151]. Архитектура сверточной нейронной сети включает входной слой, множество сверточных слоев, слои подвыборки и выходные полносвязные слои.

Основными проблемами при применении нейронных сетей являются проблема определения оптимальной для поставленной задачи структуры нейронной сети и создание обучающей выборки. В данной главе проводится разработка метода, позволяющего получить нейронную сеть и создать обучающую выборку, которые позволят решать задачу планирования пути робота.

3.2. Постановка проблемы планирования движения на базе нейронной сети глубокого обучения

Рассматривается карта плоской среды, включающая в себя известные препятствия. Пример такой среды показан на рис. 3.1. Препятствия на карте обозначены красными прямоугольниками, начальная точка обозначена черной точкой, а целевая точка – черной звездой.

При отсутствии карты среды она строится на основе данных системы технического зрения, как показано на рис. 3.2. При этом препятствия на карте достраиваются по их видимой части в соответствии с описанием, представленным в разделе 2.2 данной диссертации.

В обоих случаях отличий в процедуре обучения робота нет.

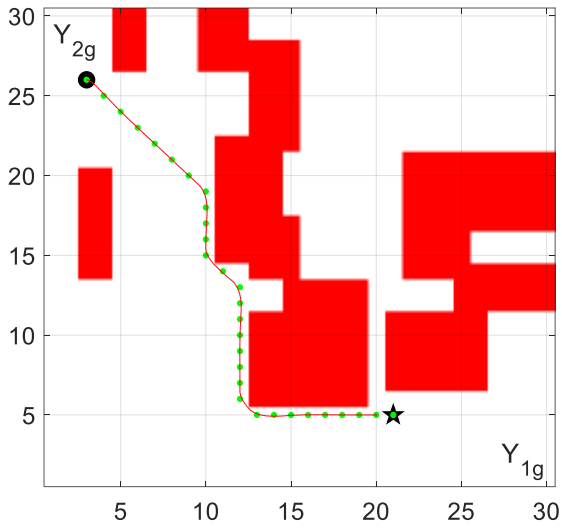


Рисунок 3.1 – Пример карты для обучения нейросетевого планировщика

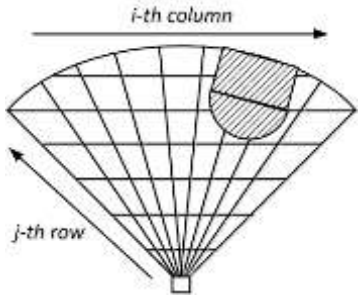


Рис. 3.2. Представление видимой части карты ячейками

В ходе диссертационного исследования в качестве учителя используется алгоритм D*, который позволяет эффективно перестраивать маршрут. При необходимости, вокруг препятствий могут создаваться области с повышенной опасностью, на которые накладываются штрафы при планировании с использованием алгоритма D*. Например, на рис. 3.3 представлен пример карты, на которой каждое препятствие окружено зонами, движение в которых штрафуется. При этом, при удалении от препятствия значение штрафов уменьшается.

Для нейронной сети карта среды подается в виде цветного изображения. Математически это означает, что на нейронную сеть подаются 3 мат-

рицы, как показано на рис. 3.4. В первой матрице задаются занятые и свободные для движения ячейки. Вторая матрица содержит текущее положение робота, а третья матрица – целевой положение.

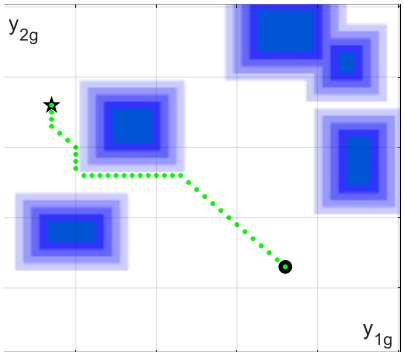


Рис. 3.3. Представление препятствий с учетом штрафов на движении непосредственно вблизи этих препятствий

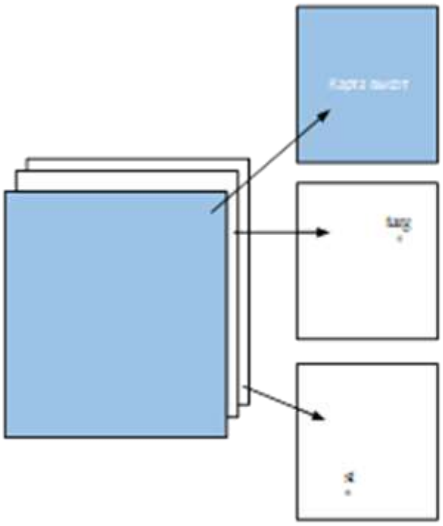


Рис. 3.4. Представление карты для нейронной сети

Задача обучения нейронной сети классифицировать подаваемое на нее изображение на девять классов, восемь из которых соответствует направлению движения, как показано на рис. 3.5, а девятый класс соответствует отсутствию пути в целевую точку. При этом направление движения определяется алгоритмом D*.

Возможная вычислительная эффективность при использовании нейронной сети базируется на двух моментах:

– при обучении на нейронную сеть подается карта среды и только текущее направление движения, вычисляемое алгоритмом D*, а не вся траектория движения;

– нейронная сеть, в силу своей структуры хорошо подходит для параллельной обработки, в отличие от итерационных процедур оптимизации, которые требуют последовательных вычислений при попятном движении от целевой точки к начальной точке.

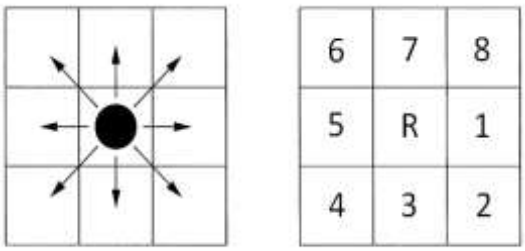


Рис. 3.5. Классификация среды по направлениям движения

3.3. Итерационное создание обучающей выборки и обучение нейронной сети

Предлагаемый метод обучения нейронной сети базируется на схеме, включающей создание предварительной обучающей выборки и итерационной процедуры создания финальной обучающей выборки [198].

Процесс создания предварительной обучающей выборки представлен на рис. 3.6.

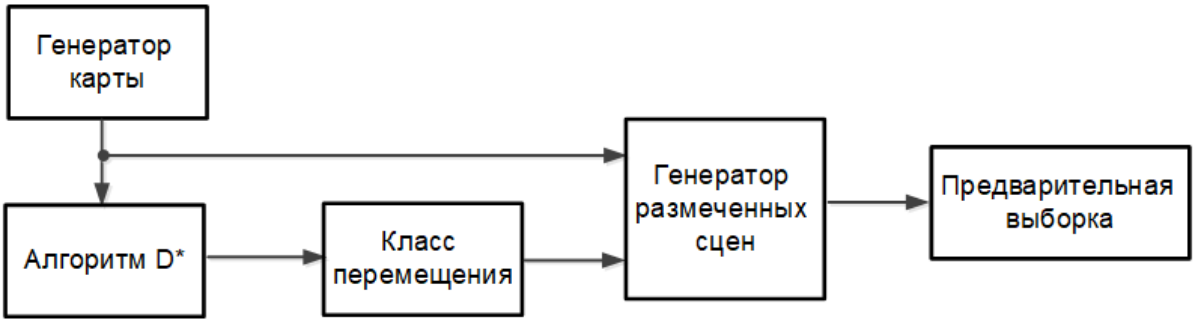


Рис. 3.6. Процесс создания предварительной выборки

Генератор карты создает моделируемую среду в виде матрицы с препятствиями, положением робота и целевой точкой. Карта поступает в алгоритм D*, который определяет класс перемещения в соответствии с секторами, показанными на рис. 3.5. Карта вместе с заданным классом перемещения поступает в генератор размеченных сцен, который создает изображения, помещаемые в предварительную выборку.

На каждой итерации выбирается часть подвыборки и осуществляется стандартный процесс обучения с учителем, представленный на рис. 3.7. В ходе обучения из подвыборки, которая является частью предварительной выборки, генерируются примеры для обучения, которые поступают в нейронную сеть. Выход нейронной сети сравнивается с классом, определенным D* и на основании полученной ошибки осуществляется настройка весовых коэффициентов нейронной сети.

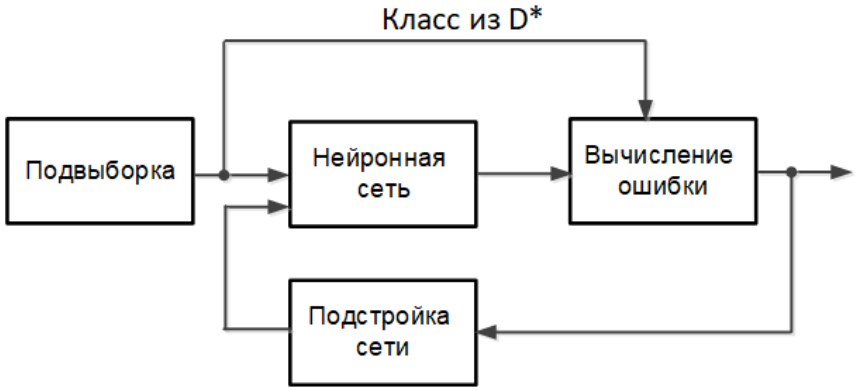


Рис. 3.7. Процесс обучения нейронной сети

В процессе первой итерации подвыборка формируется из предварительной выборки случайным образом. При этом выбранные для обучения примеры помечаются, чтобы повторно не добавляться в обучающую подвыборку на последующих итерациях. В процессе последующих итераций осуществляется дополнение обучающей подвыборки путем фильтрации с использованием обученной на предыдущей итерации нейронной сети. Данный процесс представлен на рис. 3.8.

Описанная итерационная процедура обучения описывается следующей последовательностью шагов.

1) На k-й итерации генерируется пример P_{ki}. Рассматривается сеть со структурой NN_k.

2) Каждый пример P_{ki} является помеченным, т.е. для него известно правильное решение, сгенерированное алгоритмом D*.

3) Если k>1 то пример P_{ki} подается на обученную на предыдущем шаге нейронную сеть NN_{k-1}. Если выход нейронной сети совпадает с решением, сгенерированным алгоритмом D*, то такой пример не добавляется в обучающую подвыборку. В противном случае пример P_{ki} записывается в обучающую подвыборку.

4) Проводится обучение нейронной сети NN_k в соответствии с процедурой обучения, представленной на рис. 3.7.

5) Итерации продолжаются до тех пор, пока не достигается требуемая точность обучения.

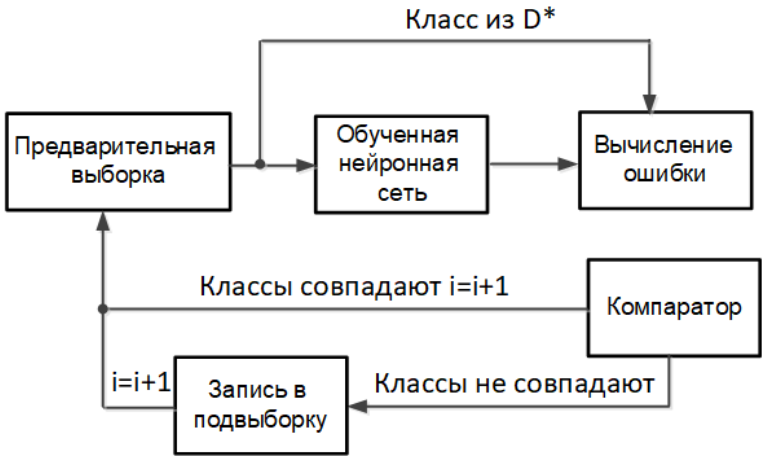


Рис. 3.8. Процесс фильтрации обучающих примеров

Данная процедура фильтрации примеров обусловлена тем фактом, что обучающая выборка должна содержать достаточное для обучения нейронной сети число обучающих примеров, которые характеризуют различные ситуации. При случайном генерировании разные ситуации могут

появляться с разной частотой. Например, для среды с заполнением препятствиями 20 – 40 %, примеры, в которых робот двигается в узких проходах, возникают примерно в 10 раз реже, чем примеры, в которых робот двигается с последовательным обходом одного-двух препятствий. Аналогично дело состоит и при формировании обучающей выборки экспериментальным методом. Поэтому при случайном формировании обучающей выборки она будет содержать большие число часто встречающихся ситуаций, и значительно меньшее число редко встречающихся ситуаций. Однако, для обучения нейронной сети необходимо некоторое достаточное для обучения сети число примеров, характеризующих ситуацию, независимо от того, насколько часто она встречается на практике.

В этой связи накопление редких ситуаций возможно либо за счет их создания при участии оператора, либо за счет значительного увеличения обучающей выборки. Автоматизированное создание образцов для обучения – это сложный, сложный и трудоемкий процесс, особенно при большом разнообразии возможных ситуаций. Он связан с большим объемом не автоматизированной работы. Контроль накопления обучающей выборки также осуществляется вручную. Случайное накопление обучающей выборки приводит к значительному увеличению обучающей выборки.

Рассмотренная на рис. 3.8 процедура позволяет в автоматическом режиме отобрать те примеры, которые необходимы для достижения заданной точности. Эффект действия описанной процедуры фильтрации представлен на рис. 3.9, на котором приведена диаграмма распределения примеров в зависимости от расстояния до препятствия для среды, заполненной препятствиями до 30 %, при объеме выборки 45 000 примеров [201].

Среднеквадратичное отклонение числа ситуаций от среднего значения, равного 9000 для случайной и фильтрованной выборки равны:

$$S_1 = \frac{\sqrt{(3.3-9)^2+(4.5-9)^2+(12.1-9)^2+(15.5-9)^2+(6.4-9)^2+(3.2-9)^2}}{6} = 2{,}0 \text{ м}, \qquad (3.1)$$

$$S_1 = \frac{\sqrt{(5.4-9)^2+(8.4-9)^2+(9.8-9)^2+(10.1-9)^2+(7.5-9)^2+(3.8-9)^2}}{6} = 1{,}22 \text{ м.} \qquad (3.2)$$

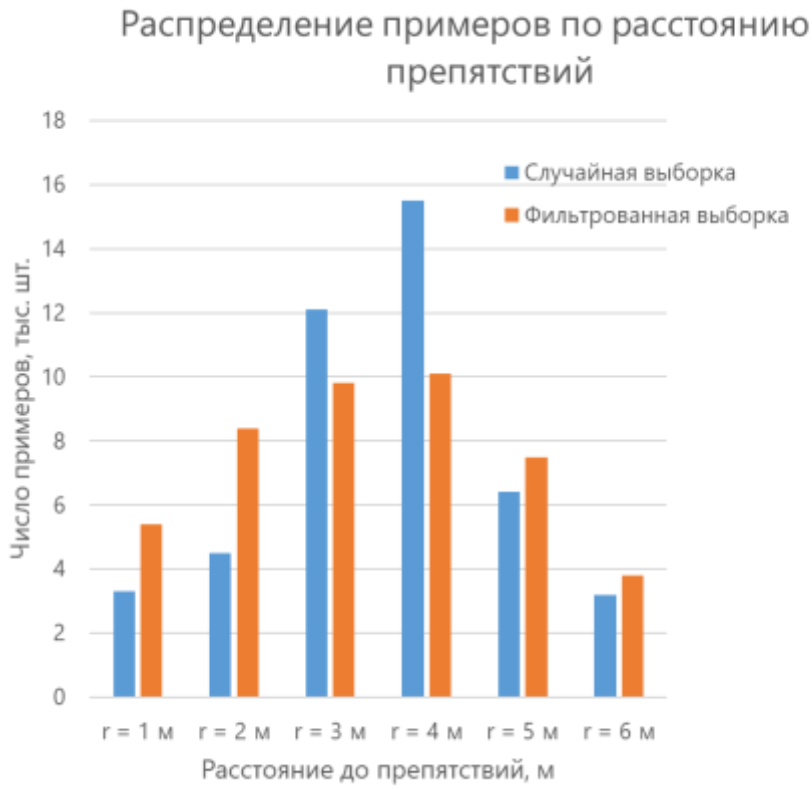


Рис. 3.9. Влияние фильтрации на распределение обучающих примеров по расстоянию до препятствий

Из рис. 3.9 хорошо видно, что фильтрация заметно изменяет распределение примеров в зависимости от расстояния до препятствия.

Аналогичный результат обнаружен и рассмотрении распределения числа примеров, в зависимости от направления движения, как следует из рис. 3.10. В данном случае такая ситуация объясняется особенностями используемого как учитель планировщика D*, который в неопределенных ситуациях отдает предпочтения движению по направлениям «Вперед», «Назад», «Влево» и «Вправо» по отношению к движению по диагоналям. Также существенное влияние оказывает тот факт, что ситуации, когда пути в целевую точку не существует, встречаются достаточно редко. Из рис. 3.10 видим, что включение процедуры фильтрации кратно увеличивает число

ситуаций, в которых робот должен двигаться вдоль диагональных направлений. Особенно сильно увеличивается число примеров, в которых нет пути из текущего положения робота в целевую точку. В случайной выборке число таких примеров равно около 200, а в фильтрованной – около 2800.

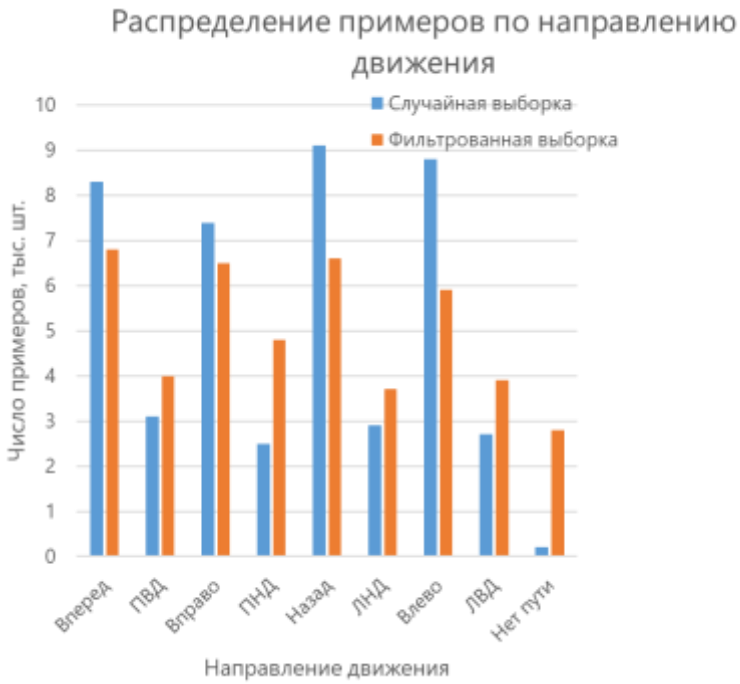


Рис. 3.10. Влияние фильтрации на распределение обучающих примеров по направлению движения

Таким образом, предложенная итерационная процедура фильтрации обучающей выборки позволяет сформировать обучающие примеры таким образом, чтобы число обучающих примеров, характеризующих различные ситуации, было достаточным. При этом вся процедура проводится автоматически.

3.4. Исследование итерационного алгоритма методом численного моделирования

Проведено численное исследование предложенного итерационного алгоритма с использованием комплекса Matlab. В ходе исследования решалась задача обучения нейронной сети планировать маршрут таким же образом, что и алгоритм «учителя», в качестве которого выбран D*.

Начальный вариант фильтрации выбран таким образом, при возникновении коллизии в фильтруемую выборку подается точка траектории, которая предшествовала коллизии, т.е. в которой нейронная сеть неправильно классифицировала ситуацию. Такой способ фильтрации оказался не эффективным, т.к. неверное решение, которое привело к коллизии, могло быть принято не непосредственно перед коллизией, а ранее. В этой связи процедура фильтрации примеров для обучения была модифицирована таким образом, чтобы при возникновении коллизии в обучающую выборку добавлялись карты с положением робота во всех точках его траектории. Это позволяет существенно повысить успешность достижения цели.

На первой итерации использована нейронная сеть NN1, включающая входной слой, 4 скрытых сверточных слоя с 16 фильтрами и 2 полносвязных слоя, содержащих 16 и 9 нейронов. Полносвязный слой с 9 нейронами является выходным слоем сети. Все скрытые слои применяют функцию активации ReLU, которая обеспечивает наиболее быстрое обучение. На первой итерации использована выборка, состоящая из 1000 изображений для каждого класса (всего 9 000 примеров). В качестве метода обучения используется стохастических градиентный поиск со сглаживанием (sgdm).

Результаты обучения нейронной сети с применением процедуры фильтрации и без нее представлены в табл. 3.1, в которой представлен объем выборки, оценка точности, полученная непосредственно при обучении и частота успешного (без коллизий) достижения цели, полученная путем моделирования движения робота с нейросетевым планировщиком. При моделировании дополнительно сгенерирована независимая выборка объемом 2000 тестовых ситуаций. При этом, на каждой карте нейронная сеть принимает решение о требуемом направлении движения несколько десятков раз, т.е. всего классифицируются десятки тысяч ситуаций. На первой итерации фильтрация выборки не производилась, поэтому в ней нет разделения точность при обычной и отфильтрованной выборках.

Таблица 3.1 Результаты обучения при точечном добавлении примеров

Итерация	***Объем выборки***	***Оценка точности, %***	***Частота успешного достижения %***
1	1000	–	–
2	4000, фильтрованная	79	50
	4000, обычная	83	43
3	10000, фильтрованная	84	79
	10000, обычная	88	70
4	12000, фильтрованная	89	85
	12000, обычная	93	75
5	13000, фильтрованная	94	92
	13000, обычная	96	80

Оценка точности работы сети на первой итерации не проводилась. На второй итерации с помощью сети NN1 произведена фильтрация выборки, в которую добавлялись только те ситуации, в которых возникали коллизии. При объем созданной выборки равен 36 000 примеров, по 9 000 примеров для каждого класса. При этом получено, что оценка точности для нейронной сети, обученной с использованием процедуры фильтрации, составила 79 %. Для сравнения, оценка точности, полученная при обучении нейронной сети с той же структурой, составила 83 %. Однако при моделировании движения выяснилось, что частота успешного достижения цели для нейронной сети, обученной с использованием фильтрации, составляет 50 %, а для

нейронной сети, обученной без фильтрации выборки – 43 %. Этот факт объясняется тем, что при оценке точности стандартными средствами обучения в Matlab используется часть обучающей выборки. Т.к. обучающие выборки в двух описанных случаях разные по сложности, то сравнивать их друг с другом не корректно. Именно для этого в качестве критерия успешности использована частота достижения цели без коллизий. При этом для проверки успешности обучения сетей использована специальная тестовая выборка, которая отличается от случайной тем, что в ней отсутствуют примеры, в которых между начальным положением робота и целью нет ни одного препятствия. Отсутствие таких ситуаций в обучающей выборке обусловлено тем, что в них не нужна нейронная сеть.

Типовой результат моделирования при планировании маршрута с использованием нейронной сети NN1 представлен на рис. 3.11, на котором зелеными точками показан маршрут, проложенный D*, черными точками – маршрут, проложенный нейронной сетью NN1, а сплошной линией – траектория движения робота. Из рис. 3.11 видим, что имеются коллизии.

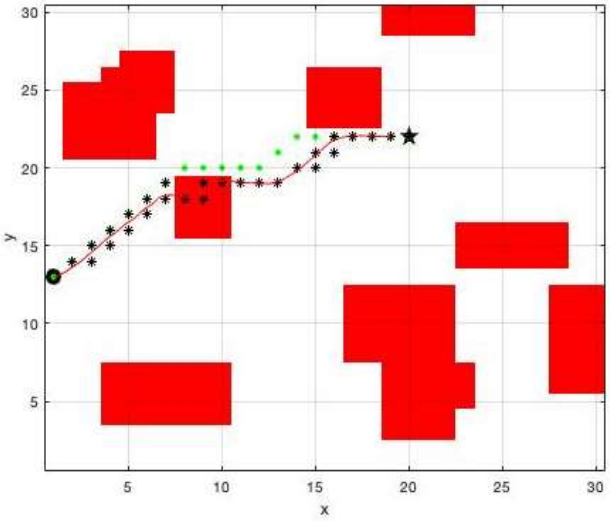


Рис. 3.11. Планирование маршрута нейронной сетью NN1

На третьей итерации обучается нейронная сеть NN2, которая включает входной слой, 5 сверточных слоев с 32 фильтрами и 3 полносвязных выходных слоя, содержащих 32, 16 и 9 нейронов. Объем выборки составил

90000 примеров – по 10000 примеров для каждого класса. Частота достижения целевой точки без коллизии составила 79 % для нейронной сети, обученной на отфильтрованной базе, и 70 % – для той же нейронной сети, но обученной на обычной выборке.

Результаты моделирования обученной сети NN2 представлены на рис. 3.12 из которого видно, что коллизии отсутствуют, однако имеются небольшие отклонения от оптимальной траектории, что приводит к увеличению ее длины.

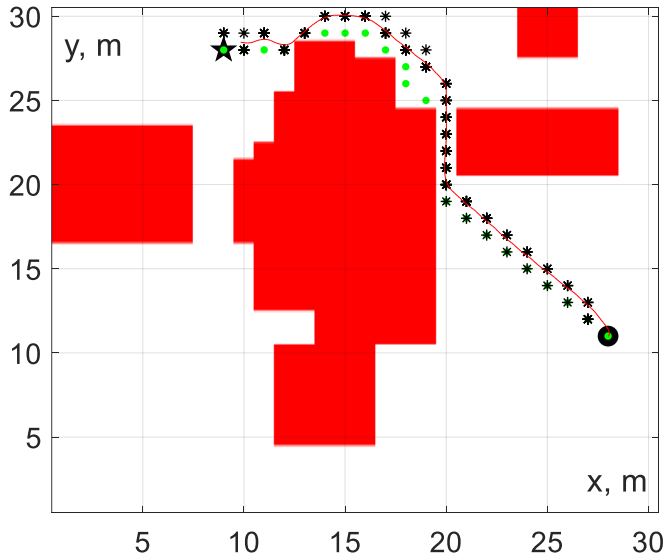


Рис. 3.12. Планирование маршрута нейронной сетью NN2

На четвертой итерации нейронная сеть NN2 использована для отбора примеров обучающей выборки. Обучаемая сеть NN3 включает входной слой, 9 скрытых сверточных слоев с 40 фильтрами и 3 полносвязных выходных слоя, содержащих 32, 16 и 8 нейронов. Частота достижения целевой точки без коллизии составила 85 % для нейронной сети, обученной на отфильтрованной базе, и 75 % – для той же нейронной сети, но обученной на обычной выборке.

На пятой итерации нейронная сеть NN3 использована для фильтрации обучающей выборки. Обучается сеть NN4, которая включает входной слой, 10 скрытых сверточных слоев с 40 фильтрами и 3 полносвязных выходных слоя, содержащих 32, 16 и 8 нейронов. Частота достижения целевой точки

без коллизии составила 92 % для нейронной сети, обученной на отфильтрованной базе, и 80 % – для той же нейронной сети, но обученной на обычной выборке.

На рис. 3.13 представлен процесс обучения нейронной сети при числе эпох обучения 16 и параметре MinBatchSize=128.

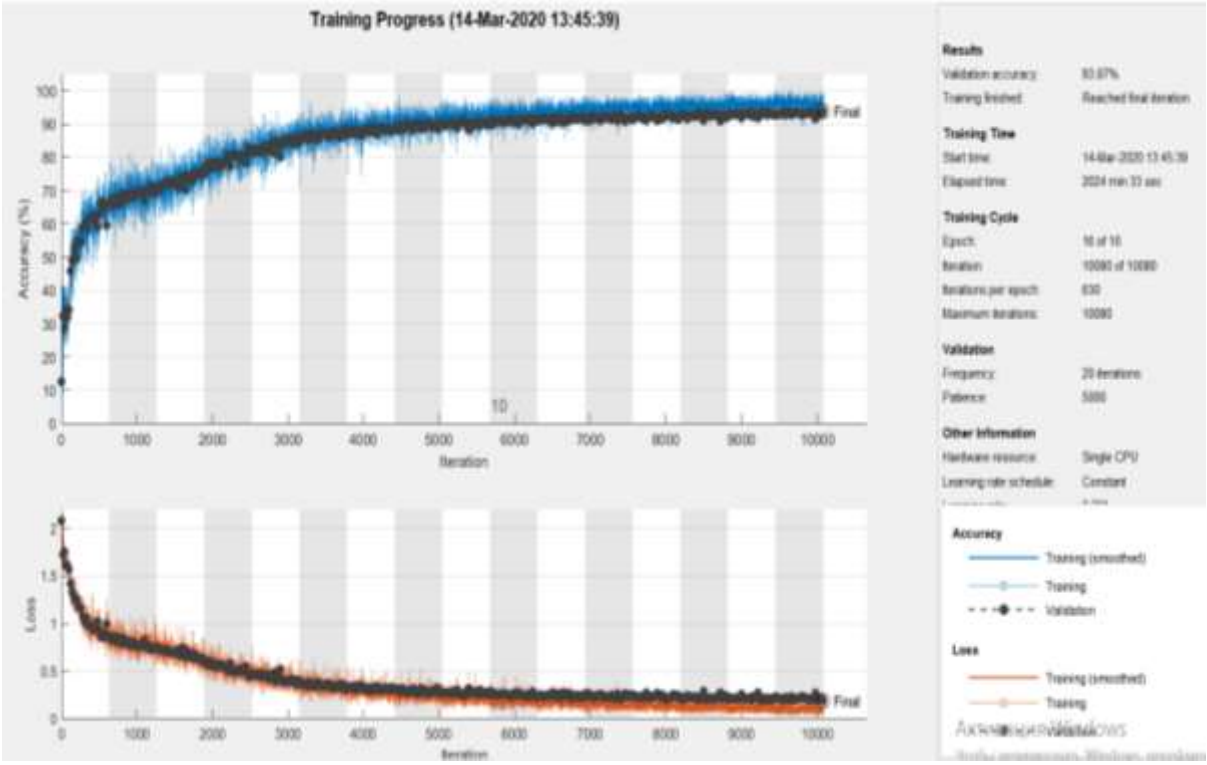


Рис. 3.13. Планирование маршрута нейронной сетью NN2

Таким образом, из табл. 3.1 видно преимущество применения фильтрации при формировании обучающей выборки, которое заключается в повышении точности принятия решений на основе нейронной сети. При этом, вычислительная сложность нейронной сети не увеличивается, т.к. изменяется только содержание обучающей выборки. Полученная в результате разница в точности составила 12 %.

На рис. 3.14 представлены результаты моделирования обученной нейронной сети NN4 в процессе планирования пути робота к целевой точке. Видно, что нейронная сеть планирует маршрут без коллизий и достаточно близко к D*. При этом отличия в траекториях не приводят к увеличению длины планируемого пути.

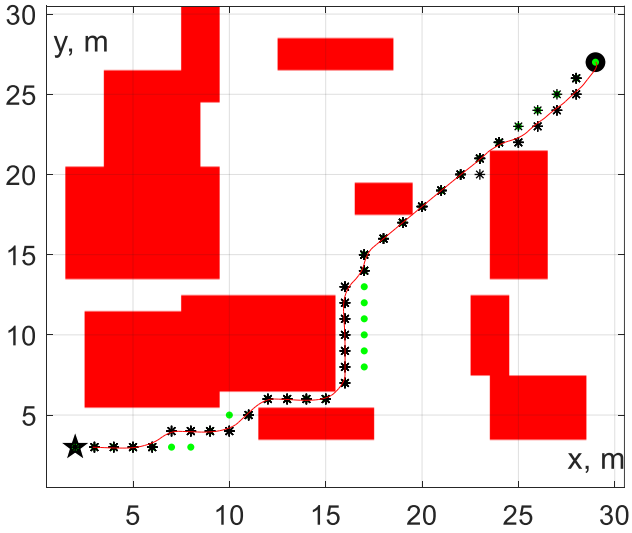


Рис. 3.14. Планирование маршрута нейронной сетью NN4

Отметим, что при отсутствии фильтрации частота успешного достижения целевой точки 92 % достигается нейронной сетью NN4 при объеме обучающей выборки 25000 примеров для каждого класса, т.е. снижение размера обучающей выборки составляет более чем в 2 раза.

Также было проведено сравнение времени расчета траектории движения с помощью нейронной сети и алгоритма D*. При этом использовался стандартный персональный компьютер с процессором Intel Core 2,9 ГГц с оперативной памятью 8 Гбайт. При использовании карты проходимости получено среднее время расчета траектории 0,025 с для нейронной сети и 0,33 с для алгоритма D*. Отметим, что данное преимущество получено без использования параллельных вычислений, которые имеют высокую эффективность в случае нейронных сетей.

3.5. Применение нейронной сети для выбора алгоритма планирования

Рассмотрим еще одну задачу, решение которой эффективно на базе нейронной сети. В силу того, что скорость мобильного робота согласно выражениям (2.10), (2.12), (2.13) уменьшается при приближении к препятствиям, то наиболее короткая траектория не всегда является оптимальной по времени движения. В этой связи в диссертации рассмотрена нейронная

сеть, осуществляющая выбор алгоритма планирования. Рассмотрено два алгоритма. Первый – это оригинальный D*, который строит маршрут минимальной длины. Второй – модифицированный D*, который строит маршрут, проходящий на некотором расстоянии от препятствий, что позволяет не сбрасывать скорость движения. Обучение нейронной сети осуществляется по схеме с учителем. Нейронная сеть получает на вход карту с положением робота, целевой точки и препятствиями. Аналогичная карта поступает на алгоритмы D* и модифицированный D*. Далее осуществляется моделирование движения робота по каждой из траекторий, в результате которой определяется время перемещения. Правильное решение назначается тому алгоритму, который обеспечивает минимальное время перемещения.

Обучение нейронной сети осуществляется с помощью итерационной процедуры, изложенной в разделе 3.3. Реализовано 4 итерации, в ходе которых создана выборка объемом 50 000 изображений. Точность обучения сети составила около 88 %.

По результатам численных исследований среднее время движения составила около 30 с. При этом применение нейронной сети, выбирающей оптимальный алгоритм, позволило сократить время движения на 4 – 6 %, что составляет 1,2 – 1,8 с.

3.6. Выводы по главе 3

В данной главе предложен метод планирования траектории робота, отличающийся использованием нейронной сети, вырабатывающей текущее направление движения. В результате обучения нейронной сети она повторяет траекторию, формируемую алгоритмов D, но без непосредственного расчета всей траектории движения. Как показали численные исследования, метод позволяет сократить время планирования по сравнению с D*, в среднем на 30 % для рассмотренной среды. Этот эффект достигается в неопределенных и динамических средах.

Также в данной главе предложен новый метод итерационного обучения нейронной сети. Данный метод позволяет на каждой последующей итерации отобрать только те ситуации, в которых нейронная принимает неверные решения. Предложенный метод позволяет уменьшить объем обучающей выборки до 40 %, а также сократить время создания обучающей выборки за счет высокой степени автоматизации.

Глава 4. Экспериментальные исследования разработанных методов и алгоритмов

4.1. Описание объекта исследования

В данной главе рассматривается робот 4-х колесный робот WM4-1A, представленные на рис. 4.1 [202, 203].



Рис. 4.1. Внешний вид колесного робота WM4-1A

Робот WM4-1A имеет 4 колеса с фиксированным углом поворота. Каждое колесо может независимо изменять скорость вращения. Мобильный робот оснащен бортовым вычислителем Jetson Nano, основные характеристики которого представлены в табл. 4.1.

Таблица 4.1 Основные характеристики процессора Jetson Nano

Видеокарта	Maxwell со 128 ядрами
Процессор	Четырехъядерный ARM A57 с частотой 1,43 ГГц
Память	4 Гб LPDDR4, 64-bit 25,6 Гбит/с
Флэш-память	microSD

Процессор Jetson Nano относится к линейке процессоров NVidia и является наименее производительным из этой линейки. В настоящее время в России имеются отечественные процессоры, превосходящие этот процессор по производительности.

Большое число ядер видеопроцессора (128) позволяет использовать данный бортовой вычислитель для приложений, связанных обработкой нейронных сетей. Процессор Jetson Nano совместим с Raspberry pi по интерфейсам. Плата процессора Jetson Nano обеспечивает возможность работы с изображениями разрешением до 3840 × 2160 пикселей.

В качестве датчиков среды робот использует камеру T265 и лидар Ydlidar X4. Основные характеристики камеры T265, производимой фирмой Intel, представлены в табл. 4.2.

Таблица 4.2 Основные характеристики камеры T265

Поддержка технологии SLAM	V-SLAM, погрешность позиционирования около 1 %, время обновления 6 мс
Процессор	Intel Movidius Myriad 2.0 VPU
Угол зрения, °	163±5°
Количество осей инерционного датчика	6
Потребляемая мощность, Вт	1,5
Габариты, мм	108 x 24.5 x 12.5
Вес, г	55

Камера T265 оснащена процессором Intel Movidius Myriad 2.0, двумя широкоугольными объективами, имеется инерционный датчик с 6-ю осями. Программное обеспечение камеры реализует технологию V-SLAM, которая решает задачу одновременного картографирования и определения положения на основе визуальной информации [194]. В камере имеется встроенный инфракрасный фильтр, который позволяет использовать ее совместно с твердотельными лидарами и камерами глубин.

Характеристики кругового лидара Ydlidar X4 представлены в табл. 4.3. Данный лидар проводит измерения с использованием метода лазерной триангуляции, который поясняется на рис. 4.2.

Таблица 4.3 Основные характеристики лидара Ydlidar X4

Диапазон действия	0,12-10 м
Разрешение диапазона	<0,1 (1% от фактического диапазона)
Разрешение угла	0,48-0,52
Скорость передачи данных	128000bps
Частота обновления информации, Гц	10

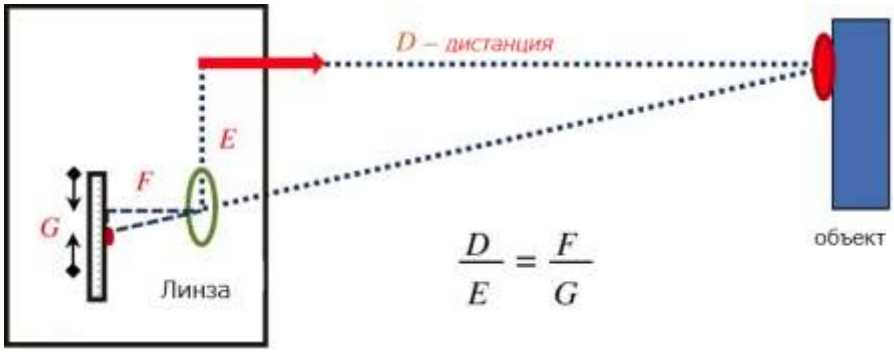


Рис. 4.2. Метод лазерной триангуляции

На рис. 4.2 величины E (расстояние между линзой и излучателем), F (расстояние до светочувствительного элемента) и G (смещение отраженного луча) известны, а расстояние до препятствия D вычисляется по формуле

$$D = E\frac{F}{G}. \quad (4.1)$$

4.2. Проведение экспериментов с использованием симулятора

Симулятор представляет собой программный комплекс, который написан на языке C++. Эксперименты проведены на персональном компьютере, оснащенном процессором Intel Core i5 11G, графической картой Intel Iris XE, ОЗУ 16 Ггб. На персональном компьютере установлена операционная система Windows10. Из этой системы с помощью подсистемы WSL2 осуществляется запуск симулятора, который работает в Ubuntu 20.04.

Для визуализации результатов исследования применяется система RVIZ2, которая является инструментом с открытым кодом. Все элементы симулятора обмениваются данными между собой в системе ROS2, что обеспечивает его мультиплатформенность. Симулятор построен таким образом, что позволяет переносить модули планирования на бортовые вычислители, на которых установлен ROS2.

Целью экспериментальных исследований является анализ времени планирования и требуемой памяти на основе накопленной в процессе симуляции статистики. Для достижения поставленной решены следующие задачи:

– разработать модель среды с различной степенью заполнения препятствиями, включая лабиринты;

– разработать программу, имитирующую модель движения колесного робота;

– разработать программное обеспечение, реализующее как созданные в диссертации алгоритмы планирования, так и известные алгоритмы;

– провести статистического моделирования и анализ результатов моделирования в созданном симуляторе.

Для реализации среды моделирования использована встроенная в ROS2 библиотека Occupancy Grid, которая описывает позволяет описывать частично картографированную среду. Данная библиотека позволяет моделировать среду в виде массива ячеек, каждая из которых содержит одно из трех значений, характеризующих состояние занята, свободна, неизвестно. Также возможно задание значения из диапазона [0 1], которое характеризует вероятность нахождения в ячейке препятствия.

Препятствия в среде генерируются в виде четырехугольников, которые могут образовывать более сложные конфигурации.

Для исследования используются следующие среды:

– среда без препятствий;

– среда заполнением препятствиями до 30 %;

– среда со множеством путей из текущего положения в целевое с заполнением препятствиями более 50 %;

– среда лабиринт с единственным путем от текущей точки к целевой с заполнением препятствиями более 50 %.

Исследуются алгоритмы A*, D*, волновой, RRT, MPN-RRT (алгоритм быстрорастущих случайных деревьев с двумя родительскими узлами), PRM, нейросетевой. Размер среды 50×50 метров, размер ячейки равен 1 м.

Во всех экспериментах использовался набор данных обеспечивающий доверительный интервал около 1,5 % от вычисляемых оценок. Доверительная вероятность принята 95 %.

В первом эксперименте исследовалась среда без препятствий. Результаты эксперимента сведены в табл. 4.4.

Таблица 4.4 Результаты исследования алгоритмов

Метод планирования	Время расчета тарек-тории, с	Требуемый объем памяти, Кб
A*	0,00261	47792
D*	0,0315	270592
Волновой	2,168	66596
RRT	0,357	64932
MPN-RRT	0,211	85334
PRM	1,777	74670
Нейросетевой	0,19	305132

Из табл. 4.4 следует, что по объему занимаемой памяти все алгоритмы, кроме D*, примерно одинаковые по требованиям. Алгоритм D* требует в несколько раз больше памяти, чем остальные. Также следует отметить несколько повышенные требования по объему памяти для алгоритма быстрорастущих случайных деревьев с двумя родительскими узлами. Это

объясняется необходимостью хранить большее количество связей между узлами дерева. Ожидаемо высокие требования к используемому объему памяти и для нейросетеового алгоритма.

По времени выполнения наиболее быстро работает в среде без препятствий A*. Остальные алгоритмы требуют на порядки большего времени расчета траектории. Можно отметить, что достаточно высокое быстродействие демонстрируют нейросетевой алгоритм и алгоритм случайных деревьев с двумя родительскими узлами. Волновой алгоритм и алгоритм вероятностной дорожной карты показали наименьшее быстродействие.

Полученные результаты хорошо объяснимы. Требующий наибольшее время для расчетов волновой алгоритм ищет на свободном пространстве соседние ячейки, на которые может распространиться волна далее. Т.к. в среде отсутствуют препятствия, то таких свободных точек много, поэтому волновой алгоритм работает медленнее других.

Также достаточно долго работает алгоритм вероятностных дорожных карт (PRM). Это связано с тем, что алгоритм генерирует достаточно большое число узлов на некотором расстоянии от текущего положения дерева.

Среднюю вычислительную эффективность показывают алгоритмы быстрорастущих случайных деревьев, быстрорастущих случайных деревьев с двумя родительскими узлами и нейросетевой алгоритм.

Следует отметить, что указанные выше преимущества и недостатки для среды без препятствий не имеют большого практического значения, т.к. любой алгоритм поиска пути обычно проверят наличие коллизий на прямой линии, соединяющей текущее положение робота и целевую точку. Такая проверка позволяет обнаруживать ситуацию, в которой целевая точка находится в прямой видимости, т.к. когда оптимальный маршрут до нее представляет прямую линию.

Вторая среда включает препятствия с максимальным заполнением 30 %. Аналогичная среда использована а главах 2 и 3. Пример такой среды

в разработанном симуляторе представлен на рис. 4.3. На данном рисунке заполненность препятствиями составляет около 15 %. Также имеется препятствие U-образной формы.

Результаты моделирования представлены в табл. 4.5.

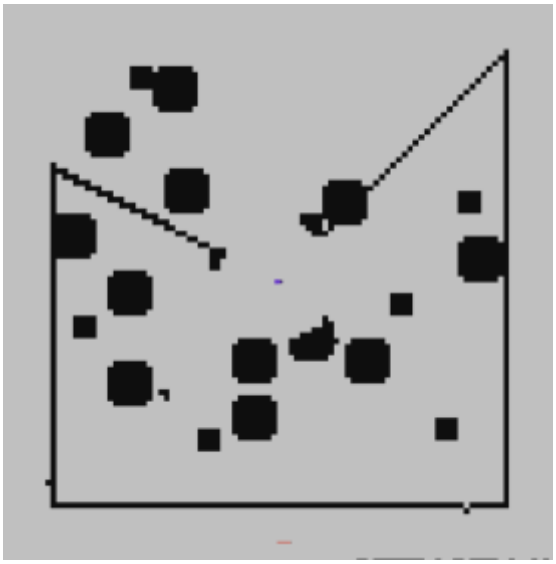


Рис. 4.3. Пример среды с препятствиями в симуляторе

Таблица 4.5 Результаты исследования алгоритмов на карте с препятствиями

Метод планирования	Время расчета тарек-тории, с	Требуемый объем па-мяти, Кб
A*	0,463	49916
D*	0,588	271036
Волновой	1,956	68184
RRT	0,361	65111
MPN-RRT	0,217	86938
PRM	9,025	74708
Нейросетевой	0,19	305132

Согласно табл. 4.5 для данной среды использование памяти алгоритмами существенно не изменилось. Вместе с тем, по времени расчета траектории наилучший результат показал нейросетевой алгоритм планирования,

разработанный в главе 3 настоящего диссертационного исследования. Это связано с тем, что он не строит всю траекторию, а только текущее направление движения.

Достаточно высокое быстродействие показал алгоритм быстрорастущих случайных деревьев с двумя родительскими узлами (MPN-RRT), разработанный в главе 2 настоящей диссертационной работы. Далее следуют оригинальный алгоритм быстрорастущих случайных деревьев, алгоритм A* и D*. Волной алгоритм по времени расчета в исследуемой среде показывает существенно худшее время расчета. Наибольшее время для планирования траектории требуется для алгоритма вероятностной дорожной карты (PRM). Это обусловлено необходимостью размещать достаточно большое число промежуточных точек, значительная часть из которых генерируется так, что между ними имеется препятствие. В этой связи процесс генерирования промежуточных точек занимает много времени.

Отметим также, что нейросетевой алгоритм занимает то же время и тот же объем памяти, что и на пустой карте. Это также связно с тем, что предложенный в главе 3 метод обучения не подразумевает построение всей траектории, а только выработку текущего направления движения.

Незначительно улучшилось время построение траектории волновым методом, что связно с уменьшением числа свободных ячеек.

Третья среда представляет собой лабиринт, представленный на рис. 4.4. Наполненность препятствиями среды составила около 52 %. Число путей из текущей точки в целевую более одного.

Результаты исследования различных алгоритмов планирования пути в данной среде представлены в табл. 4.6. Из данной таблицы следует, что наиболее эффективным с точки зрения времени планирования маршрута является алгоритм A*. Хорошие результаты также показывают алгоритм D* и волновой алгоритм. Нейросетевой алгоритм сохраняет свои показатели и

для этой среды. Наихудшие показатели времени расчета траектории дают вероятностные методы PRM, MPN-RRT и RRT.

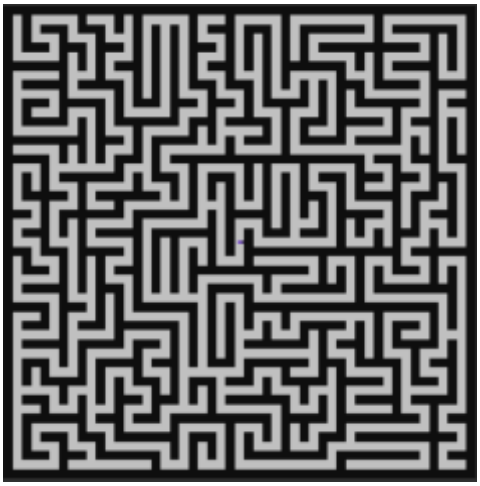


Рис. 4.4. Лабиринт с несколькими путями от начальной точки в целевую

Отметим, что данный недостаток для вероятностных методов известен и может быть уменьшен путем потенциального случайного узла в некоторой локальной области относительно текущего родительского узла. Также возможно уменьшение числа генерируемых узлов за счет снижения гладкости получаемой глобальной траектории.

Таблица 4.6 Результаты исследования алгоритмов в лабиринте с множеством путей

Метод планирования	Время расчета тарек-тории, с	Требуемый объем па-мяти, Кб
A*	0,017	45604
D*	0,032	271232
Волновой	0,04	68132
RRT	9,871	66126
MPN-RRT	5,839	87253
PRM	120,0	81004
Нейросетевой	0,19	305132

Отметим, что волновой алгоритм существенно, на несколько порядков снизил время расчета траектории в среде с большим числом препятствий.

Четвертая среда моделирования, показанная на рис. 4.5, представляет собой специфический лабиринт с единственным путем из текущего положения в целевую точку. Плотность заполнения препятствиями данной среды составляет около 50 %.

Результаты исследования различных алгоритмов планирования пути в данной среде представлены в табл. 4.7. Согласно этой таблице, наилучшие результаты по времени показали алгоритмы A*, D* и волновой алгоритм. Нейросетевой алгоритм планирования пути, как и ранее, сохранил свои показатели. Вероятностные методы показали время, аналогичное третьей среде моделирования.

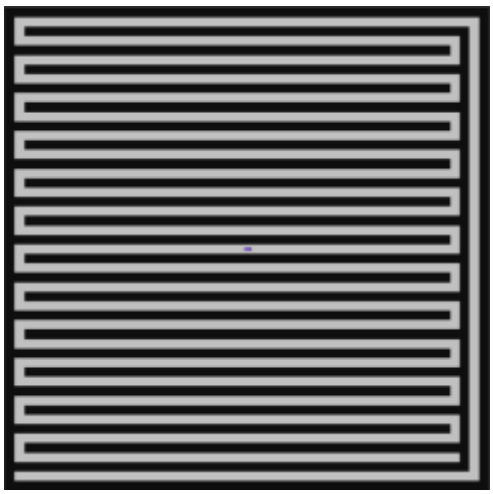


Рис. 4.5. Лабиринт с единственным путем от начальной точки в целевую

Таким образом, можно сделать выводы, что в средах с малой и средней плотностью застройки наилучшее быстродействие показывают вероятностные методы планирования пути. В средах с лабиринтами лучшее качество демонстрируют методы, основанные на клеточной декомпозиции.

По результатам моделирования можно отметить, что нейросетевой алгоритм планирования движения сохраняет свои показатели для всех вред, что важно во многих приложениях, в которых можно точно указать время

обновления маршрута. Также результаты моделирования подтвердили преимущества разработанного в главе 2 данной диссертации метода MPN-RRT по сравнению с базовым методом RRT.

Таблица 4.7 Результаты исследования алгоритмов в лабиринте с единственным путем

Метод планирования	Время расчета тарек-тории, с	Требуемый объем памяти, Кб
A*	0,026	48024
D*	0,091	269272
Волновой	0,047	68052
RRT	9,864	66115
MPN-RRT	5,714	87100
PRM	121,0	81227
Нейросетевой	0,19	305132

Можно также отметить, что метод быстрорастущих случайных деревьев с несколькими родительскими узлами имеет определенное сходство с методом дорожных карт, однако в MPN-RRT существенно ограничено число генерируемых в заданной области узлов дерева.

4.3. Проведение экспериментов с использованием колесного робота

В реальной среде возникают особенности реализации алгоритмов планирования, связанные с решением задач картографирования и навигации. Для решения этой задачи использована технология одновременной локализации и картографирования SLAM [194]. В диссертации использован итеративный алгоритм ближайших точек [195], а также адаптивный алгоритм Монте-Карло.

Эксперименты проводились в НИИ робототехники и процессов управления Южного федерального университета. Картографированная часть полигона и коридора представлены на рис. 4.6.

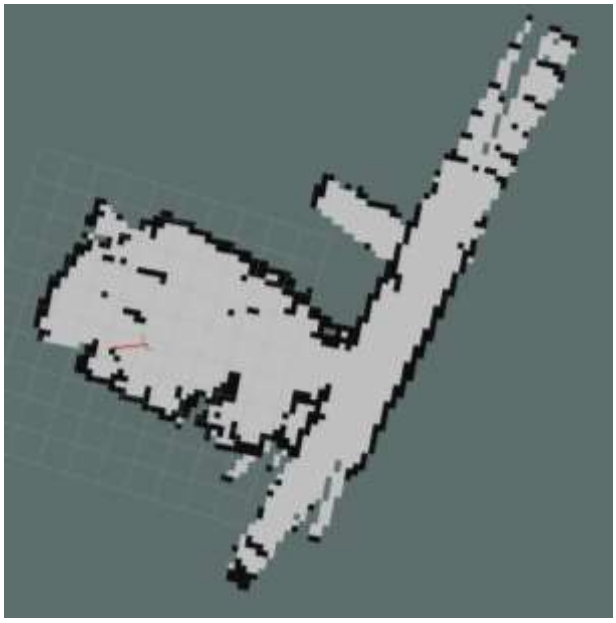


Рис. 4.6. Картографированная часть полигона

Результаты экспериментальных исследований алгоритмов планирования представлены в табл. 4.8.

Таблица 4.7 Результаты экспериментальных исследований алгоритмов планирования на полигоне НИИ робототехники и процессов управления ЮФУ

Метод планирования	Время расчета тарек-тории, с	Требуемый объем па-мяти, Кб
A*	2,568	49916
D*	3,595	271036
Волновой	5,236	68184
RRT	1,357	63525
MPN-RRT	0,711	87763
PRM	9,025	74708
Нейросетевой	0,650	314253

В целом, сравнивая табл. 4.7 и 4.5 можно сделать вывод, что результаты испытаний в симуляторе практически полностью совпадают. Наилучшие результаты по времени построения маршрута показали разработанный

в главе 2 настоящей диссертации алгоритм MPN-RRT и нейросетевой планировщик, который представлен в главе 3 настоящего диссертационного исследования.

Нейросетевой алгоритм планирования был дополнительно исследован в системе группового управления колесных роботов. В такой системе использован отдельный алгоритм для создания целевых точек, в которые должны перемещаться роботы. Нейронная сеть используется для планирования маршрута отдельного робота, а устранение коллизий обеспечивается отдельным алгоритмом.

Для планирования использована нейронная сеть из 9 сверточных слоев. Сверточные слои включают 64 фильтра с ядром 2×2. Используется функция активации ReLu. Также применяется один объединяющий слой. Перед выходных слоем используется полносвязный слой.

Время работы нейросетевого планировщика в системе управления движением составило около 0,4 с. Эксперименты проводились на полигоне НИИ робототехники и процессов управления ЮФУ. Начальное расположение роботов и отработанная ими траектория представлена на рис. 4.7.

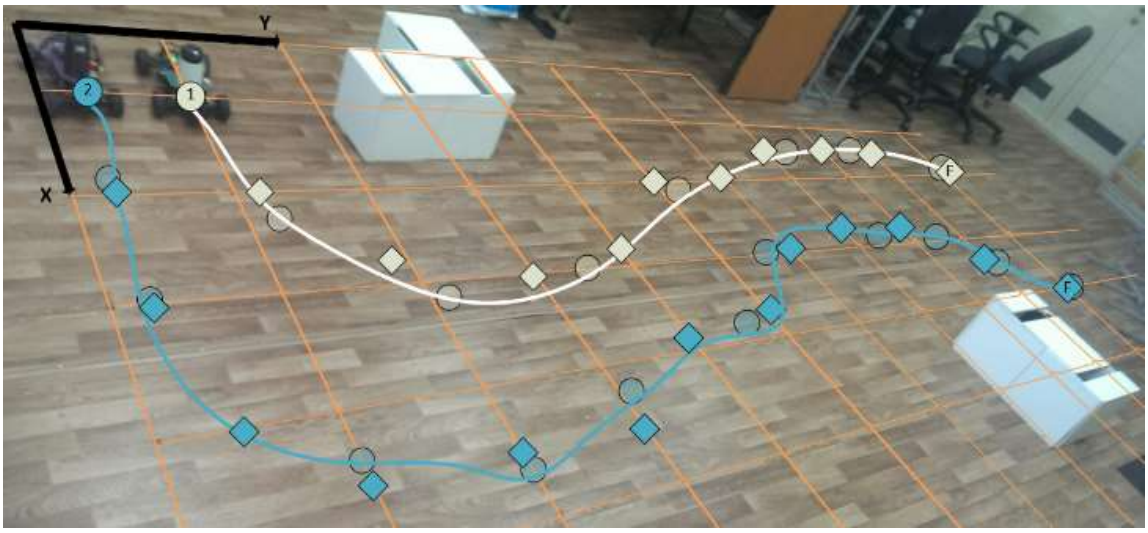


Рис. 4.7. Роботы на полигоне НИИ РиПУ ЮФУ

Средняя ошибка отработки заданных траекторий роботами составила 0,12 м при скорости движения 1 м/с.

4.4. Выводы по главе 4

В данной главе проведены численные и реальные эксперименты по сравнительному моделированию различных алгоритмов планирования пути, включая разработанные в диссертации.

По результатам проведенных экспериментальных исследований сделаны следующие выводы:

– нейросетевой алгоритм планирования, разработанный в главе 3 настоящей диссертации, обеспечивает одинаковое быстродействие и объем занимаемой памяти, вне зависимости от плотности заполнения среды препятствиями. При этом нейросетевой алгоритм показывает достаточно высокое быстродействие, но предъявляет высокие требования к занимаемой памяти. Отметим, что за счет возможностей параллельного выполнения нейросетевой обработки быстродействие планировщика может быть существенно повышено;

– разработанный в главе 2 настоящей диссертации алгоритм планирования быстрорастущих случайных деревьев с несколькими родительскими узлами показал высокое быстродействие в среде без лабиринтов с малой и средней плотностью заполнения препятствиями;

– в средах с лабиринтами возможно повышение быстродействия разработанного в главе 2 настоящей диссертации алгоритм планирования быстрорастущих случайных деревьев с несколькими родительскими узлами за счет снижения требований к гладкости траектории. Сглаживание полученной траектории возможно отдельной процедурой.

Заключение

Основным результатом диссертационной работы является решение имеющей практическую значимость актуальной научной задачи разработки, обоснования и исследования методов и алгоритмов глобального планирования движения в картографированных средах, обеспечивающих повышение быстродействия, а также учет динамических изменений среды и ограничений, накладываемых свойствами мобильных роботов.

В процессе разработки указанных методов и алгоритмов были решены следующие задачи:

– в первой главе диссертации проведен обзор и анализ известных методов планирования движения, кратко рассмотрены подходы к решению задачи планирования, с учетом динамических изменений среды и ограничений ан маневры;

– во второй главе проведены разработка и исследования двухэтапного метода планирования движения, обеспечивающего быстрое построение траектории начального приближения модифицированным алгоритмом быстрорастущих случайных деревьев и ее оптимизацию локальной итерационной процедурой. Разработанный метод позволяет учитывать ограничения на маневры без усложнения алгоритмов планирования;

– в третьей главе осуществлена разработка интеллектуального планирования движения, который вырабатывает оптимальное направление движения в текущий момент времени, что позволяет использовать его в динамических средах;

– в четвертой главе проведено экспериментальное исследование разработанных методов и алгоритмов.

Полученные в работе результаты позволяют сделать следующие обладающие новизной научные выводы:

1) Использование виртуальных препятствий, добавляемых на карту среду, позволяет учесть ограничения роботов на углы поворота мобильного

робота, зависящие от его скорости. При этом указанное расширение функциональных возможностей осуществляется без усложнения алгоритма планирования;

2) Использование двух или трех родительских узлов в алгоритме быстрорастущих случайных деревьев позволяет снизить длину планируемого пути от 6.5 до 20 % по сравнению с оригинальным алгоритмом. При одинаковой степени оптимальности траектории, которая характеризуется ее длиной, предложенный двухэтапный метод планирования позволяет сократить время расчета до 25 % по сравнению с оригинальным алгоритмом случайных деревьев. Таким образом, предложенный метод и алгоритмы планирования обеспечивают повышение качества планирования.

3) Применение интеллектуального планировщика, обучаемого с учителем, позволяет научить нейронную сеть вырабатывать текущее направление движения без явного расчета всех траектории движения до целевой точки. Это позволяет использовать нейросетевой планировщик в динамических средах.

4) В средах с лабиринтами разработанные методы могут проигрывать другим известным методам планирования, например A*. Однако в средах с малой и средней плотностью препятствий предложенные методы имеют преимущество по скорости расчета траектории движения. Кроме того, разработанный нейросетевой планировщик показывает одинаковое время расчета для всех исследованных сред.

Полученные в диссертации результаты были использованы в научно-исследовательских работах, реализованных в научно-исследовательском институте робототехники и процессов управления Южного федерального университета и ОАО «НКБ робототехники и систем управления».

Список источников

1. LaValle S.M. Planning Algorithms, *Cambridge University Press*, New York, USA. – 2006.

2. Аверкин А.Н., Гаазе-Рапопорт М.Г., Поспелов Д.А. Толковый словарь по искусственному интеллекту. – М.: Радио и связь. – 1992. – 256 с.

3. Van Den Berg J., Abbeel P., Goldberg K. LQG-MP: Optimized path planning for robots with motion uncertainty and imperfect state information. *International Journal of Robotics Research*. – 2011. – N 30(7). – Pp. 895-913.

4. Белоглазов Д.А., Гузик В.Ф., Косенко Е.Ю., Крухмалев В.А., Медведев М.Ю., Переверзев В.А., Пшихопов В.Х., Пьявченко А.О., Сапрыкин Р.В., Соловьев В.В., Финаев В.И., Чернухин Ю.В., Шаповалов И.О. Интеллектуальное планирование траекторий подвижных объектов в средах с препятствиями / Под ред. В.Х. Пшихопова. – М.: ФИЗМАТЛИТ. – 2014. – 300 с.

5. Liu S., Sun D., Zhu C. Coordinated motion planning for multiple mobile robots along designed paths with formation requirement. *Mechatronics, IEEE/ASME Transactions on*. – 2011. – N 16(6). – Pp. 1021-1031.

6. Svetlík J., Sukop M. Using of multiagent systems in robosoccer. *Acta Mechanica Slovaca*. – 2007. – N 11/2A. – Pp. 157-160.

7. Hajduk M., Sukop M. Multiagents system with dynamic box change for MiroSot. *Progress in Robotics : Communications in Computer and Information Science*. – 2009 – N 44. – Pp. 287-292.

8. Белоглазов Д.А., Гайдук А.Р., Косенко Е.Ю., Медведев М.Ю., Пшихопов В.Х., Соловьев В.В., Титов А.Е., Финаев В.И., Шаповалов И.О. Групповое управление подвижными объектами в неопределенных средах / Под ред. В.Х. Пшихопова. – М.: ФИЗМАТЛИТ. – 2015. – 304 с.

9. Каляев И.А., Гайдук А.Р., Капустян С.Г. Модели и алгоритмы коллективного управления в группах роботов, – М.: ФИЗМАТЛИТ, 2009. – 280 с.

10. Plaku E., Kavraki E.E., Vardi M.Y. Motion planning with dynamics by a synergistic combination of layers of planning. *Robotics, IEEE Transactions on*. – 2010. – N 26(3). – Pp. 469-482.

11. Пшихопов В.Х., Медведев М.Ю. Алгоритмы планирования и управления подвижными объектами в трехмерной среде с использованием неустойчивых режимов. *Известия ЮФУ. Технические науки*. – 2017. – № 2 (187). – С. 146-158.

12. Al-Sabban W.H., Gonzalez L.F., Smith R.N., Wyeth G.F. Wind-energy based path planning for electric unmanned aerial vehicles using markov decision processes. *In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. – 2012.

13. Вентцель Е.С., Овчаров Л.А. Теория вероятностей и ее инженерные приложения. – М.: Наука. – 1988.

14. Konar A., Goswami Chakraborty I., Singh S.J., Jain L.C., Nagar A.K. A Deterministic Improved Q-Learning for Path Planning of a Mobile Robot. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*. – 2013. – V. 43(5). – Pp. 1141-1153.

15. Zhu M., Otte M., Chaudhari P., Frazzoli E. Game theoretic controller synthesis for multi-robot motion planning Part I: Trajectory based algorithms. *2014 IEEE International Conference on Robotics and Automation (ICRA)*. – 2014. – Pp. 1646-1651.

16. Chen C.Y., Shih B.Y., Shih C.H.,Wang L.H. Enhancing robust and stability control of a humanoid biped robot: system identification approach. *Journal of Vibration and Control*. – 2013. – V. 19(8). – Pp. 1199-1207.

17. Вьюгин В.В. Математические основы теории машинного обучения и прогнозирования М.: 2013. – 387 с.

18. Петросян Л.А., Зенкевич Н.А., Семина Е.А. Теория игр: учебное пособие для университетов по специальности "Математика". – М.: Высшая школа. – 1998. – 304 с.

19. Красовский Н.Н. Теория управления движением. М.: Наука. – 1968. – 476 с.

20. Barkaoui M., Berger J., Boukhtouta A. An information-theoretic-based evolutionary approach for the dynamic search path planning problem. *Advanced Logistics and Transport (ICALT), 2014 IEEE International Conference on*. – 2014. – Pp. 7-12.

21. Пшихопов В.Х., Медведев М.Ю. Многоконтурное адаптивное управление подвижными объектами при решении траекторных задач. *Проблемы управления*. – 2018. – № 6. – С. 62 – 72.

22. Ulusoy A., Smith S.L., Ding X.C., Belta C. Robust multi-robot optimal path planning with temporal logic constraints. *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. – 2012. – Pp. 4693-4698.

23. Gmiterko A., Kelemen M., Virgala I., Surovec R., Vacková M. Modeling of a snake-like robot rectilinear motion and requirements for its actuators. *INES 2011: 15th IEEE International Conference on Intelligent Engineering Systems*, Poprad, Slovakia. – 2011. – Pp. 91-94.

24. Понтрягин Л.С., Болтянский В.Г., Гамкрелидзе Р.В., Мищенко Е.Ф. Математическая теория оптимальных процессов. – М.: Физматгиз. – 1961.

25. Bry A., Roy N. Rapidly-exploring random belief trees for motion planning under uncertainty. *In Robotics and Automation (ICRA), 2011 IEEE International Conference on*. – 2011.– Pp. 723-730,

26. Plaku E., Hager G.D., Sampling-based motion and symbolic action planning with geometric and differential constraints. *In Robotics and Automation (ICRA), 2010 IEEE International Conference on*. – 2010. – Pp. 5002-5008,

27. Медведев М.Ю., Костюков В.А., Пшихопов В.Х. Оптимизация движения мобильного робота на плоскости в поле конечного числа источников-репеллеров. *Труды СПИИРАН*. – 2020. – Т. 19(1). – С. 43-78.

28. Surovec R., Kelemen M., Vacková M., Virgala I. A conceptual design of the self-reconfigurable mobile robot. *ATP Journal plus*. – 2011. – Vol. 1. – Pp. 57-60.

29. Virgala I., Gmiterko A., Kelemen M., Surovec R., Prada E., Vacková M. Snake-like robot motion analysis. *ATP Journal plus*. – 2012. – Vol. 1. – Pp. 88-91.

30. Пшихопов В.Х., Медведев М.Ю. Блочный синтез робастных систем при ограничениях на управления и координаты состояния. *Мехатроника, автоматизация и управление*. – 2011. – № 1. – С. 2 – 8.

31. Wiemann T., Lingemann K., Hertzberg J. Automatic Map Creation For Environment Modelling In Robotic Simulators. *ECMS*. – 2013. – Pp. 712-718.

32. Соколов С.М., Беклемишев Н.Д., Богуславский А.А. Организация целенаправленных перемещений подвижных средств с использованием зрительных ориентиров. – 2021. – *Известия ЮФУ. Технические науки*. – № 1. – С. 235 – 246.

33. Luo R.C., Lai C.C. Enriched indoor map construction based on multisensor fusion approach for intelligent service robot. *Industrial Electronics, IEEE Transactions on*. – 2012. – V. 59(8). – Pp. 3135-3145.

34. Raja P., Pugazhenthi S. Optimal path planning of mobile robots: A review. *International Journal of Physical Sciences*. – 2012. – V. 7(9). – Pp. 1314-1320.

35. В.Х. Пшихопов, М. Ю. Медведев, Б. В. Гуренко. Алгоритмы терминального управления подвижными объектами мультикоптерного типа. *Мехатроника, автоматизация и управление*. – 2019. – Т. 20(1). – С. 44 – 51. – 10.17587/mau.20.44-51.

36. Пшихопов В.Х., Медведев М.Ю., Соловьев В.В. Гибридная система управления движением безэкипажного судна в заданную точку. *Известии ЮФУ. Технические науки*. – 2019. – № 1(203). – С. 152 – 165. – 10.23683/2311-3103-2019-1-163-176.

37. Valencia R., Andrade-Cetto J., Porta J.M. Path planning in belief space with Pose SLAM. *In Robotics and Automation (ICRA), 2011 IEEE International Conference on*. – 2011. – Pp. 78-83,

38. Sadat S.A., Chutskoff K., Jungic D., Wawerla J., Vaughan R. Feature-Rich Path Planning for Robust Navigation of MAVs with Mono-SLAM. *In Int. Conf. on Robotics and Automation (ICRA)*. – 2014.

39. Židek K., Svetlík J., Rigasová E. Environment mapping and localization on mobile robotics. *International Scientific Herald*. – 2012. – Pp. 272-280.

40. Fossel J., Hennes D., Claes D., Alers S., Tuyls K. OctoSLAM: A 3D mapping approach to situational awareness of unmanned aerial vehicles. *In Unmanned Aircraft Systems (ICUAS), 2013 International Conference on*. – 2013. – Pp. 179-188.

41. Hertzberg C., Wagner R., Birbach O., Hammer T., Frese U. Experiences in building a visual SLAM system from open source components. *In ICRA*. – 2011. – Pp. 2644-2651.

42. Ким Н.В., Носков В.П., Рубцов И.В., Аникин В.А. Автоматизация посадки беспилотного вертолета на необорудованную площадку. *Известии ЮФУ. Технические науки*. – 2020. – № 1. – С. 17 – 25.

43. Nakhaeinia D., Tang S.H., Noor S.M., Motlagh O. A review of control architectures for autonomous navigation of mobile robots. *International Journal of Physical Sciences*. – 2011. – V. 6(2). – Pp. 169-174.

44. Казаков К.А., Семенов В.А. Обзор современных методов планирования движения. *Труды ИСП РАН*. – 2016. – Т. 28(4). – С. 241-294.

45. LaValle S. Planning Algorithms. *Cambridge University Press*. – 2006. – Pp.1 – 842.

46. Lozano-Perez T. Spatial planning: a configuration space approach. *IEEE Transactions on Computers*. – 1983. – V. 32. – Pp. 108-120.

47. Сухарев А.Г. Об оптимальных стратегиях поиска экстремума. *Журнал вычислительной математики и математической физики*. – 1971. – Т. 11(4). – С. 910–924.

48. Hornung A., Wurm K.M., Bennewitz M., Stachniss C., Burgard W. OctoMap: An efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots*. – 2013. – V. 34(3). – Pp. 189–206.

49. Seda Milos. Roadmap methods vs. cell decomposition in robot motion planning. *In: Proceeding of the 6th WSEAS international conference on signal processing, robotics and automation*. World Scientific and Engineering Academy and Society (WSEAS). – 2007. – Pp. 127 – 132.

50. Regli W. Robot Lab: robot path planning. *Lecture notes of department of computer science*. Drexel University. – 2007.

51. Weigl M., Siemiaatkkowska B., Sikorski A.K., Borkowski A. Grid-based mapping for autonomous mobile robot. *Robot Autonomous Systems*. – 1993. – V. 11(1). – Pp. 13 – 21.

52. Zhu D., Latombe J.C. New heuristic algorithms for efficient hierarchical path planning. *IEEE Transactions on Robotics and Automation*. – 1991. – V. 7(1). – Pp. 9 – 20.

53. Noborio H., Naniwa T., Arimoto S. Aquadtree-basedpath-planningalgorithm for a mobile robot. *Journal of Robotic Systems*. – 1990. – V. 7(4). – Pp. 555 – 574.

54. Rosell J., Iniguez P. Path planning using harmonic functions and probabilistic cell decomposition. *IEEE International Conference on Robotics and Automation*. – 2005.

55. Cai Ch., Ferrari S. Information-driven sensor path planning by approximate cell decomposition. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*. – 2009. – V. 39(3).

56. Dugarjav B., Lee S.G., Kim D., Kim J.H., Chong N.Y. Scan matching online cell decomposition for coverage path planning in an unknown environment. *International Journal of Precise Engineering Manufacture*. – 2013. – V. 14(9). – Pp. 1551 – 1559.

57. Glavaski D., Volf M., Bonkovic M. Robot motion planning using exact cell decomposition and potential field methods. *In: Proceedings of the 9th WSEAS international conference on Simulation, modelling and optimization*. World Scientific and Engineering Academy and Society (WSEAS). – 2009.

58. Платонов А.К., Карпов И.И., Кирильченко А.А. Метод потенциалов в задаче прокладки трассы. *М.: Препринт Института прикладной математики АН СССР*. – 1974. – 27 с.

59. Платонов А.К., Кирильченко А.А., Колганов М.А. Метод потенциалов в задаче выбора пути: история и перспективы. *ИПМ им. М.В. Келдыша РАН*. – М. – 2001.

60. Khatib O. Real-time obstacle avoidance for manipulators and mobile robots. *IEEE Int. Conf. Robotics and Automation*. – 1985. – Pp. 500–505.

61. Khatib O. Real-Time Obstacles Avoidance for Manipulators and Mobile Robots, *Int. Journal of Robotics Research*. – 1986. – Vol. 5(1). – Pp. 90–98.

62. Kim J.O., Khosla P.K. Real-time obstacle avoidance using harmonic potential functions. *IEEE Transactions on Robotics and Automation*. – 1992. – V. 8(3). – pp. 338 – 349.

63. Borenstein J., Koren Y. Real-time obstacle avoidance for fast mobile robots. *IEEE Transactions on Systems, Man and Cybernetics*. – 1989. – V. 19(5). – Pp. 1179 – 1187.

64. Koditschek D.E. Task encoding: toward a scientific paradigm for robot planning and control. *Robotics and Automation systems*. – 1992. – V. 9(1–2). – Pp. 5–39.

65. Rimon E., Koditschek D.E. The construction of analytic diffeomorphisms for star worlds. *IEEE Int. Conf. Rob. and Autom*. – 1989. – V.1, – Pp. 21–26.

66. Pozna C., Precup R.-E., Koczy L.T., Ballagi A. Potential field-based approach for obstacle avoidance trajectories. *The IPSI BgD Transactions on Internet Research*. – 2002. – V. 8(2). – Pp. 40-45.

67. Padilla Castaneda M.A., Savage J., Hernandez A. and Arambula Cosio F. Local Autonomous Robot Navigation Using Potential Fields. Motion Planning. – 2008. – ISBN: 978-953-7619-01-5.

68. Adams M.D., Hu Huosheng, Probert P.J. Towards a real-time architecture for obstacle avoidance and path palnning in mobile robots. *IEEE Int. Conf. Robotics and Automation*. – 1990. – Pp. 584–589.

69. Shi P., Zhao Y. An efficient path planning algorithm for mobile robot using improved potential field. *In: IEEE International Conference on Robotics and Biomimetics*, Guilin, China. – 2009. – V. 19 – 23. – Pp. 1704 – 1708.

70. Hauge T., Brady M., Cameron S. Using moments to plan paths for the Oxford AGV. *IEEE Int. Conf. Robotics and Automation*. – 1990. – Pp. 210–215.

71. Shimoda S., Kuroda Y. and Iagnemma K. High Speed Navigation of Unmanned Ground Vehicles on Uneven Terrain Using Potential Fields. *Robotica*. – 2007. – V. 25(4). – Pp. 409 - 424.

72. Huang L. Velocity planning for a mobile robot to track a moving target-a potential field approach. *Robotics and Autonomous Systems*. – 2009. – V. 57(1). – Pp 55 – 63.

73. Borenstein J., Koren Y. Real-time Obstacle Avoidance for Fast Mobile Robots In Cluttered Enviroments. *IEEE Int. Conf. Robotics and Automation*. – 1990. – Pp. 572–577.

74. Uusitalo T., Johansson S.J. A Reactive Multi-agent Approach to Car Driving using Artificial Potential Fields. *In Proc. 2011 IEEE Conference on Computational Intelligence and Games*. – 2011. – Pp. 203 – 210.

75. Ge S.S., Cui Y.J. New Potential Functions for Mobile Robot Path Planning. *IEEE Transactions on Robotics and Automation*. – 2000. –V. 16(5). – Pp. 615 – 620.

76. Sfeir J., Saad M., Saliah-Hassane H. An improved artificial potential field ap- proaches to real-time mobile robot path planning in an unknown environ- ment. *In: IEEE international symposium on robotic and sensors environments*, Canada. – 2011. – V. 17–18. – Pp. 208 – 213.

77. Soloviev V.V., Pshikhopov V.K., Shapovalov I.O., Finaev V.I., Beloglazov D.A. Planning of the mobile robot motion in non-deterministic environments with potential fields method. *International Journal of Applied Engineering Research*. – 2015 – V. 10. – Pp. 41954-41961.

78. Beloglazov D.A., Finaev V.I., Titov A.E., Shapovalov I.O., and Soloviev V.V. Group robot control in non-deterministic environments using the potential field method.*16th International Conference on Control, Automation and Systems (ICCAS 2016)*. – 2016.

79. Hart P.E., Nilsson N.J., Raphael B.A., Formal Basis for the Heuristic Determination of Minimum Cost Paths, *IEEE Transactions on Systems Science and Cybernetics*. – 1968. – V. 2. – Pp. 100 – 107.

80. Stentz A., Optimal and efficient path planning for partially known environments, *In Intelligent Unmanned Ground Vehicles*, Springer, Boston, MA, USA. – 1997. – Pp. 203–220.

81. Korf R.E., Depth-first iterative-deepening. An optimal admissible tree search, *Artificial Intelligence*. – 1985. – V. 27(1). – Pp. 97–109.

82. Zhou R., Hansen E.A., Memory-Bounded {A*} Graph Search, *The Florida AI Research Society Conference - FLAIRS*. – 2002. – Pp. 203–209.

83. Holte R., Perez M., Zimmer R., MacDonald A., Hierarchical A*: Searching abstraction hierarchies efficiently, *Proceedings of the thirteenth national conference on Artificial intelligence*. – 1996. – V. 1. Pp. 530–535.

84. Koenig S., Likhachev M., Furcy D., Lifelong Planning A*. *Artificial Intelligence*. – 2004. – V. 155(1-2). – Pp. 93–146.

85. Chen B.Y., Chen X.-W., Chen H.-P., Lam W.H.K., Efficient algorithm for finding k shortest paths based on re-optimization technique, *Transportation Research Part E: Logistics and Transportation Review*. – 2020. – V. 133. – Article number 101819.

86. Zhang X., Wylie B., Oscar C., Moore C.A., Time-Optimal and Collision-Free Path Planning for Dual-Manipulator 3D Printer, *IEEE Transactions on Systems, Man, and Cybernetics: Systems*. – 2020. – 9283493. – Pp. 2389-2396.

87. Elbanhawi M., Simic M., Sampling-based robot motion planning: A review, *IEEE Access*. 2014. – V. 2. – Pp. 56-77.

88. Лавренов Р.О. Планирование маршрута для беспилотного наземного робота с учетом множества критериев оптимизации / Р.О. Лавренов, И.М. Афанасьев, Е.А. Магид // *Третий Всероссийский научно-практический семинар "Беспилотные транспортные средства с элементами искусственного интеллекта"*: Труды семинара. – М: Перо. – 2016. – С. 10-20.

89. Kavraki L.E., Svestka P., Latombe J.-C., Overmars M.H., Probabilistic roadmaps for path planning in high-dimensional configuration spaces, *IEEE Transactions on Robotics and Automation*. – 1996. – V. 12 (4). – Pp. 566-580,

90. Chakravorty S., Kumar S., Generalized Sampling-Based Motion Planners, *IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics*. – 2011. – V. 41(3).

91. Ravankar A., Ravankar Ab., Emaru T., Kobayashi Y., HPPRM: Hybrid Potential Based Probabilistic Roadmap Algorithm for Improved Dynamic Path Planning of Mobile Robots, *IEEE Acsess*. – 2020. – V. 8. – Pp. 221743 – 221766.

92. Щербаков В.С., Корытов М.С. Оптимизация трассы автомобильной дороги на рельефе с препятствиями методом вероятностной дорожной карты. *Вестник Сибирской государственной автомобильно-дорожной академии*. – 2012. – Вып. 6(28). – С. 88 – 92.

93. Лю В. Методы планирования пути в среде с препятствиями (обзор). *Математика и математическое моделирование*. – 2018. – № 1. – С. 15 – 49.

94. LaValle S.M., Kuffner J. Randomized kinodynamic planning, *Int. Journal of Robotics Research*. – 2001. – V. 20(5). – Pp. 378–400.

95. LaValle S.M., Kuffner J.J. Rapidly-exploring random trees: Progress and prospects, *2000 Workshop on the Algorithmic Foundations of Robotics*. – 2000. – Pp. 293–308.

96. Wang X., Li X., Guan Y., Song J., Wang R. Bidirectional Potential Guided RRT* for Motion Planning, *IEEE Acsess*. – 2019. – V. 7. – Pp. 95046 – 95057.

97. Галемов Р.Т., Масальский Г.Б. Планирование траектории манипулятора для движущейся цели. *Кибернетика и программирование*. – 2018. – № 2. – С. 9 – 28.

98. Qureshi A., Ayaz Y. Potential functions based sampling heuristic for optimal path planning, *Autonomous Robot*. – 2016. – V. 40. – Pp 1079–1093.

99. Karaman S., Frazzoli E. Sampling-based algorithms for optimal motion planning, *The International Journal of Robotics Research*. – 2011. – 30(7). – Pp. 846–894.

100. Пшихопов В.Х. Аттракторы и репеллеры в конструировании систем управления подвижными объектами. *Известия ТРТУ*. – 2006. – № 3(58). – С. 117–123.

101. Пшихопов В.Х. Организация репеллеров при движении мобильных роботов в среде с препятствиями. *Мехатроника, автоматизация, управление*. – 2008. – № 2. – С. 34–41.

102. Пшихопов В.Х., Медведев М.Ю. Групповое управление движением мобильных роботов в неопределенной среде с использованием неустойчивых режимов. *Труды СПИИРАН*. – 2018. – Вып. 60. – С. 39-63.

103. Pshikhopov V., Medvedev M. Mobile Objects Control in Three-Dimensional Area Using the Hybrid Decentralized Algorithm. *5th International Conference on Control, Decision and Information Technologies*, CoDIT. – 2018. – Pp. 452-457.

104. Chen L., Shan Y., Tian W., Li B., and Cao D. A Fast and Efficient Double-Tree RRT∗-Like Sampling-Based Planner Applying on Mobile Robotic Systems, *IEEE/ASME Transactions on Mechatronics*. – 2018. – V. 23(6). – Pp. 2568 – 2578.

105. Wang J., Meng M.Q.-H., Khatib O. EB-RRT: Optimal Motion Planning for Mobile Robots, *IEEE Transactions on Automation Science and Engineering*. – 2020. – V. 17(4). – Pp. 2063-2073.

106. Wang R., Zhang X., Fang Y., Li B. Virtual-Goal-Guided RRT for Visual Servoing of Mobile Robots With FOV Constraint, *IEEE Transactions on Systems, Man, and Cybernetics: Systems*. – 2021.

107. Janos J., Vonasek V., Penicka R. Multi-Goal Path Planning Using Multiple Random Trees, *IEEE Robotics and Automation Letter*. – 2021. –V. 6(2). – Pp. 4201-4208.

108. Spitz S.N. and Requicha A.A.G. Multiple-goals path planning for coordinate measuring machines, *in Proc. IEEE Int. Conf. Robot. Automat.* – 2000. – V. 3. – Pp. 2322–2327.

109. Devaurs D., Siméon T., and Cortés J. A multi-tree extension of the transition-based RRT: Application to ordering-and-pathfinding problems in continuous cost spaces, *in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.* – 2014. – Pp. 2991–2996.

110. Vonásek V. and P˘eni˘cka R. Space-filling forest for multi-goal path planning, *in Proc. 24th IEEE Int. Conf. Emerg. Technol. Factory Automat.* – 2019. – Pp. 1587–1590.

111. Lulu L, Elnagar A. A comparative study between visibility-based roadmap path planning algorithms. *In: 2005 IEEE/RSJ international conference on intelligent robots and systems*. – 2005. – Pp. 3263 – 3271.

112. De Berg M., Cheong O., Van Kreveld M., Overmars M. Computational Geometry: Algorithms and Applications, *3rd Edition. Springer-Verlag*. – 2008. – 386 p.

113. Takahashi O, Schilling RJ. Motion planning in a plane using generalized Voronoi diagrams. *IEEE Trans Robot Autom*. – 1989. – V.5(2). – Pp. 143 – 150.

114. Bremermann H.J. The evolution of intelligence. The Nervous system as a model of its environment, *Technical Report no.1, contract no. 477(17)*. Washington, Seattle: Dept. Mathematics Univ. – 1958.

115. Shibata T., Fukuda T. Robot motion planning by genetic algorithm with fuzzy critic. *In: Proc. 8th IEEE international symp. On intelligent control*. – 1993. – Pp. 565 – 570.

116. Shing M.T., Parkar G.B. Genetic algorithm for the development of real-time multi-heuristic search strategies. *In: Proc. 5th conf. On genetic algorithm*. – 1993. – Pp. 565 – 570.

117. Xia J., Michalewicz Z., Zhang L., Trojanowski K. Adaptive evolutionary planner/ navigator for mobile robot. *IEEE Transcation on Evolutionary Computation*. – Apr 1997. – V.1(1).

118. Kang X., Yue Y., Li D., Maple C. Genetic algorithm based solution to dead problems in robot navigation. *Int J Comput Appl Technol*. – 2011. – V.41(3-4). – Pp. 177 – 184.

119. Даринцев О.В., Мигранов А.Б. Система планирования движения группы мобильных микроботов на основе генетических алгоритмов. *Известия российской академии наук. Теория и системы управления*. – 2007. – № 3. С. 163 – 173.

120. Shi P., Cui Y. Dynamic path planning for mobile robot based on genetic al- gorithm in unknown environment. *In: Proceedings of the ChineseControl and decision Conference*. – 2010. – Pp. 4325 – 4329.

121. Pratihar D.K., Deb K., Ghosh A. Fuzzy-Genetic algorithm and time-optimal obstacle free path generation for mobile robots. *Eng Optim*. – 1999. – V. 32(1). – Pp. 117 – 142.

122. Hui N.B., Pratihar D.K. A comparative study on some navigation schemes of a real robot tackling moving obstacles. *Robot Comput Integrated Manuf*. – 2009. – V. 25. – Pp. 810 – 828.

123. Wang X., Shi Y., Ding D., Gu X. Double global optimum genetic algorithm- particle swarm optimization-based welding robot path planning. *Eng Optim*. – 2016. – V. 48(2). – Pp. 299 – 316.

124. Kala R. Coordination in navigation of multiple mobile robots. *Cybern Syst*. – 2014. – V. 45(1). – Pp. 1 – 24.

125. Liu F., Liang S., Xian X. Optimal robot path planning for multiple goals visiting based on Tailored genetic algorithm. *Int J Comput Intell Syst*. – 2014. – V. 7(6). – Pp. 1109 – 1122.

126. Yang S.X., Hu Y. A knowledge based GA path planning of multiple mobile robots indynamic environments. *In: 2006 IEEE conference on Robotics, Automation and Mechatronics*. – 2006. – Pp. 1 – 6.

127. Hong Q., Ke X., Alexander T. An improved genetic algorithm with co- evolutionary strategy for global path planning of multiple mobile robots. *Neurocomputing*. – 2013. – V. 120. – Pp. 509 – 517.

128. Jianjun N., Kang W., Haohao H., Liuying W., Chengming L. Robot path planning based on an improved genetic algorithm with variable length chromosome. *In: 12th international conference on natural computation, fuzzy systems and knowledge discovery (ICNC-FSKD)*. – 2016. – https://doi.org/10.1109/FSKD.2016.7603165.

129. Chen J.W., et al. Research on fuzzy control of path tracking for underwater vehicle based on genetic algorithm optimization. *Ocean Eng*. – 2018. – V. 156. – Pp. 217 – 223.

130. Lin K.P., Hung K.C. An efficient fuzzy weighted average algorithm for the military UAV selecting under group decision-making. *Knowl Base Syst*. – 2011. – V. 24(6). – Pp. 877 – 889.

131. Roberge Vincent, Tarbouchi Mohammed, Labonte Gilles. Fast genetic algo- rithm path planner for fixed-wing military UAV using GPU. *IEEE Trans Aerosp Electron Syst*. – 2018. – https://doi.org/10.1109/TAES.2018.2807558.

132. Creaser PA, Stacey BA, White BA. Evolutionary generation of fuzzy guidance laws. no. 455. *In: UKACC international conference on Control'98*. – 1998. – V. II. – p. 883 – 888.

133. Kumar A, Kumar Priyadarshi Biplab, Parhi Dayal R. Intelligent navigation of humanoids in cluttered environments using regression analysis and genetic algorithm. *Arabian J Sci Eng*. – 2018. – Pp. 1 – 24.

134. Zadeh LA. Fuzzy sets. *Inf Control*. – 1965. – V. 8(3). – Pp. 338 – 353.

135. Tang Sai Hong, Danial Nakhaeinia and Babak Karasfi Application of Fuzzy Logic in Mobile Robot Navigation. University Putra Malaysia, Malaysia // Fuzzy Logic - Controls, Concepts, Theories and Applications Source: InTech. – 2005. – Pp. 22-36

136. Brooks R.A. A Robust Layered Control System for a Mobile Robot. *IEEE Journal of Robotics and Automation*. 1896. – V. 2(1). - Pp. 14 – 23.

137. Brooks R.A. A Robot that Walks; Emergent Behavior from a Carefully Evolved Network. *IEEE International Conference on Robotics and Automation*, Scottsdale, AZ. – 1989. - Pp. 292 – 296.

138. Arkin R.C. Motor schema-based mobile robot navigation. *Int. J. of Robotic Research*. – 1989. – Vol 8. - Pp. 92 - 112.

139. Arkin R.C. & Balch T. AuRA: Principles and Practice in Review. *Journal of Experimental and Theoretical Artificial Intelligence*. – 1997. – V. 9(2/3). – 1997. – Pp. 175 - 188.

140. Yang S.X., Moallem M. & Patel R.V. A Layered Goal-Oriented Fuzzy Motion Planning Strategy for Mobile Robot Navigation. *IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics*. – 2005. – V. 35(6). – Pp. 1214 - 1224.

141. Rosenblatt J. DAMN: A Distributed Architecture for Mobile Navigation, Ph.D. Dissertation, Carnegie Mellon University Robotics Institute Technical Report CMU-RITR-97-01, Pittsburgh, PA. – 1995.

142. Saffiotti A. The uses of fuzzy logic for autonomous robot navigation: a catalogue raisonn'e, *Soft Computing Research journal*. – 1997. – V.1(4). – Pp. 180 - 197.

143. Tunstel E., Lippincott T. & Jamshidi M. Behavior Hierarchy for Autonomous Mobile Robots: Fuzzy-behavior modulation and evolution//International Journal of Intelligent Automation and Soft Computing, Special Issue: Autonomous Control Engineering at NASA ACE Center. – 1997. – V. 3(1). – Pp. 37 - 49.

144. Castellano G, Attolico G, Distante A. Automatic generation of fuzzy rules for reactive robot controllers. *Robot Autonom Syst*. – 1997. – V. 22(2) – Pp. 133 - 149.

145. Abiyev R, Ibrahim D, Erin B. Navigation of mobile robot in presence of obstacles. *Adv Eng Software*. – 2010. – V. 41(10). – Pp. 1179 – 1186.

146. Ge SS, Lai X, Mamun A. Sensor-based path planning for nonholonomic mobile robots subject to dynamic environment. *Robot Autonom Syst*. – 2007. – V.55(7). – Pp. 513 – 526.

147. Motlagh O, Hong TS, Ismail N. Development of new minimum avoidance systems for a behavior-based mobile robot. *Fuzzy Sets Syst*. – 2009. – V. 160. – Pp. 1929 – 1946.

148. Huq R, Mann G, Gosine R. Mobile robot navigation using motor schema and fuzzy context dependent behavior modulation. *Appl Soft Comput*. – 2008. – V. 8 – Pp. 422 – 436.

149. Moustris GP, Tzafestas SG. Switching fuzzy tracking control for mobile robots under curvature constraints. *Contr Eng Pract*. – 2011. – V. 19. – Pp. 45-53.

150. Madan M. Gupta, Liang Jin, Noriyasu Homma. Static and Dynamic Neural Networks: From Fundamentals to Advanced Theory. *John Wiley & Sons*, New Jersey. – 2003.

151. LeCun Ya., Yoshua B., and Geoffrey H. Deep learning. *Nature*. – 2015. – V. 521.7553. – Pp. 436-444.

152. Yu J., Ji J., Miao Z., Zhou J. Neural network-based region reaching formation control for multi-robot systems in obstacle environment. *Neurocomputing*. – 2019. –V. 333. – Pp. 11-21.

153. Geng M., Xu K., Zhou X., Ding B., Wang H., Zhang L. Learning to cooperate via an attention-based communication neural network in decentralized multi-robot exploration. *Entropy*. – 2019. – V. 21(3).

154. Agrawal P., Agrawal H. Adaptive algorithm design for cooperative hunting in multi-robots. *International Journal of Intelligent Systems and Applications*. – 2018. – V. 10 (12). – Pp. 47-55.

155. Price E., Lawless G., Ludwig R., Martinovic I., Bulthoff H.H., Black M.J., Ahmad A. Deep Neural Network-Based Cooperative Visual Tracking Through Multiple Micro Aerial Vehicles. *IEEE Robotics and Automation Letters*. – 2018. – V. 3(4). – Pp. 3193-3200.

156. Martínez-García E.A., Torres-Córdoba R., Carrillo-Saucedo V.M., López-González E. Neural control and coordination of decentralized transportation robots. *Proceedings of the Institution of Mechanical Engineers. Part I: Journal of Systems and Control Engineering*. – 2018. – V. 232(5). – Pp. 519-540.

157. Wang Y., Cheng L., Hou Z.-G., Yu J., Tan M. Optimal Formation of Multirobot Systems Based on a Recurrent Neural Network. *IEEE Transactions on Neural Networks and Learning Systems*. – 2016 – V. 27(2). – Pp. 322-333.

158. Pshikhopov, V., Medvedev, M., Vasileva, M. Neural network control system of motion of the robot in the environment with obstacles. *Lecture Notes in Computer Science*. – 2019. –V 11606. – Pp. 173 – 181.

159. Janglova D. Neural networks in mobile robot motion. *Int J Adv Robot Syst*. – 2004. – V. 1. – Pp. 15 – 22.

160. Li QL, Song Y, Hou ZG. Neural network based Fast SLAM for autonomous robots in unknown environments. *Neurocomputing*. – 2015. – V. 165. – Pp. 99 – 110.

161. Na YK, Oh SY. Hybrid control for autonomous mobile robot navigation using neural network based behavior modules and environment classification. *Aut Robots*. – 2003. – V. 15. – Pp. 193 – 206.

162. Pothal JK, Parhi DR. Navigation of multiple robots in a highly clutter terrains using adaptive neuro-fuzzy inference system. *Robotics and Automation*. – 2015. – V. 72. – Pp. 48 – 58.

163. Abu Baker A. A novel mobile robot navigation system using neuro-fuzzy rule-based optimization technique. *Res J Appl Sci Eng Technol*. – 2012. – V. 4(15). – Pp. 2577 – 2583.

164. Qiao J, Fan R, Han H, Ruan X. Q-learning based on dynamical structures neural network for robot navigation in unknown environment. *Advances in Neural Network*. – 2009. – V. 5553. Pp. 188 – 196.

165. Medvedev, M. Development of the Neural-Based Navigation System for a Ground-Based Mobile Robot / M. Medvedev, D. Brosalin, A. Kadhim // 2021 7th International Conference on Mechatronics and Robotics Engineering, ICMRE 2021, Budapest, Hungary. – 2021. – P. 35-40. – DOI 10.1109/ICMRE51691.2021.9384825.

166. Medvedev M., Pshikhopov V. Path Planning of Mobile Robot Group Based on Neural Networks. *Lecture Notes in Artificial Intelligence*. – 2020 – Pp. 51–62.

167. Neural Network Based Control System for Robots Group Operating in 2-d Uncertain Environment / A. R. Gaiduk, O. V. Martjanov, M. Yu. Medvedev, V. Kh. Pshikhopov, N. Hamdan, A. Farhood // Mechatronics, Automation, Control. – 2020. – Vol. 21(8). – P. 470-479. – DOI 10.17587/mau.21.470-479.

168. Pal PK, Kar A. Sonar-based mobile robot navigation through supervised learning on a neural net. *Aut Robots*. – 1996. – V. 3. – Pp. 334 – 355.

169. Medina-Santiago A, Campus-Anzueto JL, Vazquez-Feijoo JA, Hernandez-de- Leon HR, Mota-Grajales R. Neural control systems in obstacle avoidance in mobile robots using ultrasonic sensors. *J Appl Res Technol*. – 2014. – V. 2. – Pp. 104 – 110.

170. Syed UA, Kunwar F, Iqbal M. Guided autowave pulse coupled neural network (GAPCNN) based real time path planning and an obstacle avoidance scheme for mobile robots. *Robot Autonom Syst*. – 2014. – V. 62. – Pp. 474-486.

171. Sun C, He W, Ge W, Chang C. Adaptive neural network control of biped robots. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*. – 2017. – V. 47(2). – Pp. 315-326.

172. Sun C, He W, Hong J. Neural network control of a flexible robotic manipu- lator using the lumped spring-mass model. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*. – 2018. – V. 47(8). – Pp. 1863-1874.

173. Zhu D, Tian C, Sun B, Luo C. Complete coverage path planning of autonomous underwater vehicle based on GBNN algorithm. *J Intell Robot Syst*. – 2018. – Pp. 1-13.

174. Zhang C, Hu H, Wang J. An adaptive neural network approach to the tracking control of micro aerial vehicles in constrained space. *Int J Syst Sci*. – 2017. – V. 48(1). – Pp. 84-94.

175. Macro D. Ant colony system: a Cooperative learning approach to the trav- elling salesman problem. *IEEE transaction on evolutionary computation*. – 1997. – V. 1(1). – 53-66.

176. Guan-Zheng T, HE Huan, Aaron S. Ant colony system algorithm for real time globally optimal path planning of mobile robots. *Acta Autom Sin*. – 2007. – V. 33(3). – Pp. 279-285.

177. Кажаров А.А., Курейчик В.М. Муравьиные алгоритмы для решения транспортных задач. *Известия РАН. Теория и системы управления*. – 2010. – № 1. – С. 32-45

178. Shi Y. Particle swarm optimization: Developments, applications and resources. International Conference on Evolutional Computations. – 2001. – V. 1. – Pp. 81–86.

179. Duan H., Qiao P. Pigeon-inspired optimization: A new swarm intelligence optimizer for air robot path planning. *International Journal of Intelligent Computing and Cybernetics*. – 2014. – V. 7(1). – Pp. 24-37.

180. Radmanesha M., Kumar M., Sarim M. Grey wolf optimization based sense and avoid algorithm in a Bayesian framework for multiple UAV path planning in an uncertain environment. *Aerospace Science and Technology*. – 2018. – V. 77. – Pp. 168–179.

181. Пшихопов В.Х., Медведев М.Ю. Многоконтурное адаптивное управление подвижными объектами при решении траекторных задач. *Проблемы управления*. – 2018. – № 6. – С. 62 – 72.

182. Yang S.M., Lin Y.A. Development of an improved rapidly exploring random trees algorithm for static obstacle avoidance in autonomous vehicles. *Sensors*. – 2021. – V. 21(6). Pp. 22 – 44.

183. P. Pharpatara, B. Herisse, Y. Bestaoui, “3-D Trajectory Planning of Aerial Vehicles Using RRT”, 2017, *IEEE Transactions on Control Systems Technology*. – 2017. – V. 25(3). Pp. 1116-1123.

184. Dijkstra E.W. I note on two problem in connection with graph. *Numerishe Mathematik*. – 1959. – V. 1. – p. 269.

185. Кристофидес Н. Теория графов. Алгоритмический подход. – М.: Мир. – 1978.

186. M. Dorigo, L.M. Gambardella, “Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem,” *IEEE Transactions on Evolutionary Computation*. – 1997. – V. 1(1). Pp. 53-66.

187. V. Kostjukov, M. Medvedev, V. Pshikhopov, “Method for Optimizing of Mobile Robot Trajectory in Repeller Sources Field,” *Informatics and Automation*. – 2021. – V. 20(3). Pp. 690-726.

188. Дж. Х. Уилкинсон. Алгебраическая проблема собственных значений. М: Гл. ред ф.-мат. лит. – 1970.

189. Николенко С., Кадурин А., Архангельская Е. Глубокое обучение. Погружение в мир нейронных сетей. СПб: Изд-во Питер. – 2018.

190. Hyansu B., Gidong K., Jonguk K., Dianwei Q., Sukgyu L. Multi-Robot Path Planning Method Using Reinforcement Learning, *Applied Sciences*. – 2019. – V. 9(15). – Pp. 30 – 57.

191. Визильтер Ю.В., Вишняков Б.В., Выголов О.В., Горбацевич В.С., Князь В.А., “Технологии интеллектуальной обработки информации для задач навигации и управления беспилотными летательными аппаратами”, *Тр. СПИИРАН*. – 2016. – V. 45. – С. 26–44.

192. Redmon J., Divvala S., Farhadi A. “You Only Look Once, Unified, Real-Time Object Detection,” *Computer Vision Pattern Recognition*. – 2015.

193. Белоруцкий Р.Ю., Житник С.В. Распознавание речи на основе сверточных нейронных сетей. *Вопросы радиоэлектроники*. – 2019. – № 4. – С.47 – 52.

194. Savaria D.T., Balasubramanian R. V-SLAM: Vision-based simultaneous localization and map building for an autonomous mobile robot. *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*. – 2010.

195. Zhang Zh. Iterative Point Matching for Registration of Free-Form Curves and Surfaces. International Journal of Computer Vision. – 1994. – V. 13(2). – Pp. 119-152.

196. Фархуд, А. К. Управление мобильным роботом с применением нейронной сети для планирования движения в некартографированной среде с препятствиями / А. К. Фархуд // Известия ЮФУ. Технические науки. – 2021. – № 6(223). – С. 99-114. – DOI 10.18522/2311-3103-2021-6-99-114.

197. Trajectory Planning Algorithms in Two-Dimensional Environment with Obstacles / V. Pshikhopov, M. Medvedev, V. Kostjukov, F. Houssein, A. Kadhim // Informatics and Automation. – 2022. – Vol. 21, No. 3. – P. 459-492. – DOI 10.15622/ia.21.3.1.

198. Фархуд, А. К. Проектирование архитектуры сверточной нейронной сети (CNN-2D) для навигации мобильных роботов / Фархуд Азхар Кадим // Компьютерные и информационные технологии в науке, инженерии и управлении "КомТех-2020" : материалы Всероссийской научно-технической конференции с международным участием: в 2 т. Т. 2. – Ростов-на-Дону; Таганрог : Издательство Южного федерального университета, 2020. – С. 48-56.

199. Фархуд, А. К. Алгоритмы планирования и управления движением мобильного робота в двумерной стационарной среде с препятствиями / А. К. Фархуд // Инновационные направления интеграции науки, образования и производства: сборник материалов III Международной научно-практической конференции, Керчь, 11-15 мая 2022 года. – Керчь: Керченский государственный морской технологический университет, 2022. – С. 103-107.

200. Фархуд, А. К. Разработка планирования и управления движением мобильного робота / А. К. Фархуд // Технические науки: проблемы и решения: сборник статей по материалам LX международной научно-практической конференции. Т. 5 (55). – Москва: Интернаука, 2022. – С. 20-23.

201. Фархуд, А. К. Алгоритмы системы управления движением мобильного робота применяется нейронная сеть в среде с препятствиями / Фархуд Азхар Кадим // Научная инициатива иностранных студентов и аспирантов : сборник докладов II Международной научно-практической конференции, Томск, 26-28 апреля 2022 г. – Томск : Изд-во Томского политехнического университета, 2022. – С. 596-600.

202. Medvedev, M. Yu. The study numerical to detect collisions using neural network with D* algorithm / M. Yu. Medvedev, A. K. Farhood // Транспорт и транспортные системы: конструирование, эксплуатация, технологии : сборник научных статей / Белорусский национальный технический университет ; редкол.: С. В. Харитончик (гл. ред.) [и др.]. – Минск: БНТУ, 2022. – Вып. 4. – С. 114-119. – Режим доступа: https://rep.bntu.by/handle/data/125766 (дата обращения 27.02.2022).

203. Farhood, A. K. Mobile robot path planning in environmental based on convolution neural network algorithm / Azhar Kadhim Farhood // Компьютерные и информационные технологии в науке, инженерии и управлении "КомТех-2022": материалы Всероссийской научно-технической конференции с международным участием: в 2 т. Т. 2. – Ростов-на-Дону; Таганрог: Издательство Южного федерального университета, 2022. – С. 155-162.

Приложения

УТВЕРЖДАЮ
Генеральный директор АО НКБ
робототехники и систем управления
к.т.н., доцент
_____ Е.Ю. Косенко
«10» октября 2022 г.

АКТ

внедрения результатов диссертации аспиранта кафедры электротехники и мехатроники института радиотехнических систем и управления Южного федерального университета Фархуд Азхар Кадим Фархуд «Методы и алгоритмы планирования траекторий движения роботов в двумерных картографированных средах» на соискание ученой степени кандидата технических наук при разработке технологий планирования траекторий движения мобильных роботов.

Комиссия в составе

председатель	к.т.н., с.н.с. АО НКБ робототехники и систем управления Гуренко Б.В.;
члены:	к.т.н., с.н.с. АО НКБ робототехники и систем управления Шевченко В.А.;
	к.т.н., с.н.с. АО НКБ робототехники и систем управления Костюков В.А.

подготовила настоящий акт о том, что при разработке методов планирования движения мобильных роботов в рамках выполнения гранта РНФ 22-29-00533 «Развитие методов планирования траекторий подвижных объектов в неопределенной динамической среде» использованы следующие научные результаты диссертационной работы аспиранта кафедры электротехники и мехатроники института радиотехнических систем и управления Южного федерального университета Фархуд Азхар Кадим Фархуд «Методы и алгоритмы планирования траекторий движения роботов в двумерных картографированных средах»:

1. Двухэтапный метод планирования траектории движения робота, отличающийся использованием модифицированного алгоритма быстро растущих случайных деревьев для получения траектории первого приближения, и итерационного алгоритма локальной оптимизации.

2. Модифицированный алгоритм быстро растущих случайных деревьев, отличающийся использованием нескольких родительских узлов.

3. Метод учета ограничений на маневры робота, зависящие от его скорости, отличающийся использованием виртуальных препятствий, которые перекрывают недопустимый для движения сектор.

Использование предложенным методов и созданных на их основе алгоритмов при разработке методов планирования траекторий подвижных

объектов в неопределенной динамической среде позволило снизить время планирования оптимальной траектории движения и учесть динамические свойства мобильного робота без усложнения алгоритмов планирования, ориентированных на использование карты среды.

Анализ эффективности применения полученных Фархуд Азхар Кадим Фархуд результатов выявил снижение времени расчетов траектории движения на 25 – 30 % по сравнению с используемыми ранее алгоритмам планирования.

Председатель: _____ (подпись) к.т.н., с.н.с. АО НКБ робототехники и систем управления Гуренко Б.В.;

Члены: _____ (подпись) к.т.н., с.н.с. АО НКБ робототехники и систем управления Шевченко В.А.;

_____ (подпись) к.т.н., с.н.с. АО НКБ робототехники и систем управления Костюков В.А.

УТВЕРЖДАЮ
Директор института радиотехнических
систем и управления
Южного федерального университета
к.т.н., доцент
_____А.С. Болдырев
_____2022 г.

АКТ

внедрения результатов диссертации аспиранта кафедры электротехники и мехатроники института радиотехнических систем и управления Южного федерального университета Фархуд Азхар Кадим Фархуд «Методы и алгоритмы планирования траекторий движения роботов в двумерных картографированных средах» на соискание ученой степени кандидата технических наук в учебный процесс.

Учебно-методическая комиссия в составе

председатель	к.т.н., доцент кафедры электротехники и мехатроники института радиотехнических систем и управления Южного федерального университета Пивнев В.В.;
члены:	к.т.н., доцент кафедры электротехники и мехатроники института радиотехнических систем и управления Южного федерального университета Кульченко А.Е.;
	д.ф-м.н., профессор кафедры электротехники и мехатроники института радиотехнических систем и управления Южного федерального университета Матвеев А.И.

подготовила настоящий акт о том, что в учебном процессе кафедры электротехники используются следующие результаты диссертационной работы Фархуд Азхар Кадим Фархуд «Методы и алгоритмы планирования траекторий движения роботов в двумерных картографированных средах»:

1. Метод планирования траекторий, отличающийся использованием нейронной сети, вырабатывающей текущее направление движения, что позволяет нейронной сети повторять траекторию, формируемую учителем, но без непосредственного расчета всей траектории движения.

2. Метод обучения нейронной сети, отличающийся итерационной фильтрацией, которая отбирает те ситуации, в которых нейронная принимает неверные решения.

Указанные методы используются в лекционных материалах и практических работах дисциплины «Адаптивное управление и методы искусственного интеллекта в робототехнике», который читается для магистрантов 1-го курса направления 15.04.06 – Мехатроника и робототехника.

Материалы диссертации также использованы при подготовке магистерских диссертаций направления 15.04.06 – Мехатроника и робототехника, в том числе в диссертации Васильевой М.А. «Исследование и разработка алгоритмов планирования пути подвижных объектов» и диссертации Бросалина Д.О. «Исследование глобальных алгоритмов планирования маршрута в 2D средах по известной карте».

Председатель:  (подпись) к.т.н., доцент кафедры электротехники и мехатроники института радиотехнических систем и управления Южного федерального университета Пивнев В.В.;

Члены:  (подпись) к.т.н., доцент кафедры электротехники и мехатроники института радиотехнических систем и управления Южного федерального университета Кульченко А.Е.;

 (подпись) д.ф-м.н., профессор кафедры электротехники и мехатроники института радиотехнических систем и управления Южного федерального университета Матвеев А.И.