ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ УЧРЕЖДЕНИЕ
ФЕДЕРАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ЦЕНТР
«ИНФОРМАТИКА И УПРАВЛЕНИЕ» РОССИЙСКОЙ АКАДЕМИИ НАУК

На правах рукописи

ШМАЛЬКО Елизавета Юрьевна

Принцип синтезированного оптимального управления в робототехнических системах

Специальность 2.3.1. – «Системный анализ, управление и обработка информации, статистика»

ДИССЕРТАЦИЯ

на соискание ученой степени доктора технических наук

Научный консультант:
доктор технических наук, профессор
Дивеев А.И.

Москва, 2024

ОГЛАВЛЕНИЕ

Введение

Актуальность темы исследования. При создании новых автономных роботов и систем управления ими разработчики стремятся создавать оптимальные системы, то есть являющиеся лучшими по заданному критерию. Стремление к оптимальности является естественным в любой области. Еще Леонард Эйлер говорил о том, что оптимальность есть принцип построения Вселенной: «Поскольку строение вселенной совершенно и создано мудрым Творцом, во вселенной не возникает ничего, в чем нельзя было бы увидеть смысл какого-то максимума или минимума».

Для этих целей, начиная с 60х годов XX века, была сформирована мощная теория оптимального управления. В рамках теории была сформулирована основная задача оптимального управления как задача нахождения закона управления для заданной системы, обеспечивающего выполнение определенного критерия оптимальности.

В статье [1] Л.С. Понтрягин говорит о том, что «если не все, то во всяком случае многие математики должны в своей работе обращаться к первоисточникам, то есть к приложениям математики. Это необходимо как для того, чтобы оправдать свое существование, так и для того, чтобы влить новую свежую струю в научные исследования». Понтрягин подчеркивает важность прикладного характера математических исследований в области управления.

Оглядываясь сегодня назад, можно сказать, что эволюция в области управления была вызвана тремя основными потребностями:

- необходимостью иметь дело со все более сложными системами,
- необходимостью выполнения все более жестких требований как к самой системе, так и скорости ее создания,
- необходимостью достижения этих требований с меньшими предварительными знаниями об объекте и его окружении - то есть необходимость управления в условиях повышенной неопределенности.

Стремительное развитие робототехники предъявляет эти вызовы к разработчикам систем управления для роботов, требуя ускорения процессов создания систем автоматического управления ими.

Робототехника – сегодня одно из наиболее перспективных и бурно развивающихся направлений. В последние годы она неизменно включена в приоритетные направления научно-технического развития нашей страны: стратегии научно-технологического развития Российской Федерации, утвержденной Указом Президента Российской Федерации:

от 1 декабря 2016 г. № 642 «Переход к передовым цифровым, интеллектуальным производственным технологиям, роботизированным системам, новым материалам и способам конструирования, создание систем обработки больших объемов данных, машинного обучения и искусственного интеллекта»,

от 18 июня 2024 г. № 529 «Интеллектуальные транспортные и телекоммуникационные системы, включая автономные транспортные средства».

Количество и разнообразие роботов стремительно растет, растет и количество решаемых ими задач. В связи с большим разнообразием роботов, возможностью их быстрого создания и выведения на рынок, все очевиднее становится необходимость автоматизации подходов к созданию систем управления такими объектами.

При этом важно сохранить основную идею, что все разрабатываемые системы управления автономными робототехническими объектами должны быть не просто разработаны, а должны быть оптимальными согласно предъявляемому к ним критерию качества.

В существующей на сегодняшний день практике расчет оптимального управления сложными робототехническими системами производится предварительно на основе их математической модели. При этом модель представляет собой упрощенную версию самого описания динамики объекта, охватывающую только лишь его основные существенные свойства. Это

обусловлено рядом причин. С одной стороны, подробные динамические модели, которые точно описывают поведение роботов для некоторых систем, могут быть недоступны или их невозможно построить. С другой стороны, наиболее распространенные алгоритмы управления были разработаны с использованием упрощенных линейных моделей. Но производительность этих моделей может быть плохой, поскольку динамика робота может быть далека от линейной. В результате при переносе полученных расчетных оптимальных управлений на реальный объект возникают отклонения реального объекта от расчетной модели. В этом заключается основная проблема реализации получаемых оптимальных управлений. При решении задачи оптимального управления в классической постановке находится функция управления, как функция времени, и считается, что дальнейшая задача состоит в создании системы управления для движения объекта вблизи полученной оптимальной программной траектории. Здесь возникают проблемы определения величины отклонения от программной траектории, определения компонент вектора управления, которые отрабатывают данные отклонения объекта от траектории, определения величины управляющего воздействия по величине отклонения для уменьшения ошибки движения объекта вблизи программной траектории. Для этого на практике в реальные системы вводятся дополнительные системы стабилизации и контуры управления. Ранее и до настоящего времени этот процесс построения системы стабилизации движения выполняется вручную. Специалисты в области управления, изучая объект и заданную траекторию, определяют каналы управления, обеспечивающие определенные движения объекта в пространстве состояний, вставляют в эти каналы регуляторы и настраивают коэффициенты регуляторов для качественного движения объекта вблизи заданной траектории. Чаще всего в качестве регуляторов используются либо ПИ (пропорционально-интегральный), либо ПИД (пропорционально-интегрально-дифференцирующий) регуляторы не зависимо от модели объекта управления или вида траектории.

Теперь согласно требованиям времени необходимо автоматизировать этот процесс, то есть создать численный метод и написать программу, которая для любого объекта управления и заданной траектории любого вида определяет каналы управления и преобразует величины отклонения от заданной траектории в управляющие воздействия определенной величины и обеспечивает качественное движение объекта управления по оптимальной траектории. Такая программа в общем случае должна искать математическое выражение функции управления, т.е. структуру функции и ее параметры.

Сегодня появилась алгоритмическая конструкция, которая обеспечивает оптимизационный поиск структуры и параметров функции – это символьная регрессия. Методы символьной регрессии появились в конце двадцатого века и предназначались для решения задачи автоматического написания программ, при этом программы представлялись в виде набора функций, поэтому первоначально методы символьной регрессии реализовывались на языке функционального программирования LISP. В результате любая полученная программа методом символьной регрессии является сложной функцией. Все методы символьной регрессии кодируют искомую функцию на основе алфавита элементарных функций в форме специального кода и осуществляют поиск оптимального решения согласно заданному критерию специальным генетическим алгоритмом, в котором операции скрещивания и мутации построены так, чтобы получать новые правильные коды математических выражений функции.

Здесь следует отметить, что методы символьной регрессии – это достаточно новый малоизученный инструмент, который в настоящий период проходит апробацию для решения различных задач, где в качестве решения необходимо получить математическое выражение функции. При этом следует учитывать, что для задач управления необдуманное применение методов символьной регрессии может привести к нереализуемым математическим моделям. Поэтому в настоящей работе методы символьной регрессии используются на одном из этапов реализации предложенного принципа синтезированного управления.

Специалистами в области управления давно было замечено, что достаточным условием реализуемости модели является ее устойчивость относительно точки равновесия в области управления пространства состояний. Сегодня согласно этому условию часто управление робототехническими устройствами выполняется следующим образом. Независимо от решаемой задачи объект управления, робот, делается устойчивым относительно некоторой точки в пространстве состояний, при этом система управления, обеспечивающая его устойчивость включает некоторые параметры, которые могут изменить в положение этой устойчивой точки равновесия в пространстве состояний. Далее устойчивые точки равновесия размещаются пространстве состояний так, чтобы объект управления при последовательном переключении положения точки устойчивого равновесия решал поставленную задачу управления.

Следует отметить, что введение дополнительных контуров управления изменяет модель объекта управления. В результате, расчет оптимальности производился для одной модели, а по факту управляющие сигналы применяются для другой модели. Заметим также тот факт, что при таком подходе, вводя систему стабилизации, разработчиками вообще не ставится вопрос об оптимальности или сохранении расчетной оптимальности полученных управлений, анализируется только устойчивость.

В результате, сложность самой постановки задачи оптимального управления в робототехнике в виду нелинейности динамических объектов, их высокой размерности, необходимости учета различных фазовых ограничений, и дополнительная трудность, состоящая в реализации полученных оптимальных управлений, приводят к тому, что проблема оптимального управления в ее традиционной математической постановке все меньше и меньше решается при создании новых автономных робототехнических устройств. Сегодня мы наблюдаем, что математические исследования задач оптимального управления все больше отходят от практики их реализации.

Очевиден разрыв между теорией оптимального управления и ее прикладной реализацией, наблюдается уход от основных постулатов, выдвигаемых самим основателем теории оптимального управления Л.С. Понтрягиным.

Настоящее диссертационное исследование посвящено разработке и обоснованию нового подхода к решению **научно-технической проблемы** получения реализуемых на практике оптимальных управлений робототехническими системами в автоматизированном режиме с помощью методов машинного обучения управлению.

В работе предложен новый принцип синтезированного оптимального управления. Согласно толковому словарю, принцип – это основное исходное положение, которым следует руководствоваться в какой-либо деятельности. Предлагаемый принцип формулирует такое положение при разработке реализуемых оптимальных систем управления с возможностью применения современных численных методов машинного обучения.

Машинное обучение базируется на идее, что вычислительные системы способны демонстрировать поведение, которое не было в них явно запрограммировано, они могут выявлять закономерности или функциональные зависимости и самостоятельно вырабатывать решения. Во многих научных дисциплинах, если не во всех, основная задача исследований состоит в нахождении функциональной зависимости между определенными значениями параметров, характеризующих свойства исследуемого объекта. Если удается искомую функциональную зависимость представить в виде математической формулы, то очень часто такая формула становится законом в данной области. Важность задачи поиска математического выражения некоторой функции подтверждается и мировыми тенденциями. С помощью методов машинного обучения искомые функциональные зависимости могут быть выражены неявно, как в случае с очень популярными сегодня нейронными сетями, или же в явном виде в случае применения таких методов машинного обучения, как символьная регрессия. Представление функциональной зависимости в интерпретируемом

для человека виде имеет очевидные преимущества с точки зрения анализа получаемых решений, но методы символьной регрессии имеют целый ряд и других преимуществ применительно к разработке законов управления, поэтому активно применяются и исследуются в диссертационном исследовании.

Оперируя терминами теории оптимального управления, можно сказать, машинное обучение управления направлено на поиск закона управления некоторым объектом для оптимального достижения поставленных целей в терминах некоторого сформулированного критерия. Закон управления в общем случае представляет собой многомерную вектор-функцию, которую необходимо определить. Функция может быть задана с точностью до параметров. В общем случае, должны быть найдены как структура функции, так и ее параметры.

Сейчас большинство систем управления роботами все еще создают вручную. Основываясь на своем опыте, разработчик задает структуру системы управления, определяет каналы управления, типы регуляторов, и далее настраивает параметры заданной системы так, чтобы они удовлетворяли определенным требованиям. А ведь эту задачу можно и нужно рассматривать как задачу оптимального управления, определяя не только параметры, но и структуру системы управления оптимально.

Современные системы управления для роботов являются цифровыми и представляют собой компьютерные программы, реализованные на бортовых процессорах. Но сегодня эти программы все еще пишутся вручную программистами. Очевидно, что программы, которые должны управлять роботами для сложных задач управления, могут включать в себя несколько десятков или сотен тысяч строк. И эти программы будут увеличиваться при усложнении задач и усложнении роботов. Из сказанного следует, что ручное создание системы управления роботами - направление неперспективное, существует объективная необходимость автоматизировать этот процесс. И современные вычислительные мощности этому способствуют.

Потребность в более эффективном управлении все более сложными динамическими системами в условиях значительной неопределенности привела

к переоценке традиционных методов управления и сделала совершенно очевидной потребность в новых машинных методах теории управления. Рост возможностей машинного обучения, и в частности, таких методов эволюционного машинного обучения, как методы символьной регрессии, приводит к смене парадигмы управления, когда контроллеры обучаются, и структура и параметры искомых регуляторов ищутся как решение задачи оптимизации на нечисловом пространстве структур, открывая широкие возможности для получения интеллектуальных контроллеров, способных использовать нелинейности системы для повышения эффективности управления.

Диссертационная работа направлена на решение актуальной научно-технической проблемы автоматизированного получения реализуемых функций управления.

Цель диссертационного исследования - разработка подхода и методов для решения проблемы автоматизированного получения реализуемых законов управления в робототехнических системах на основе применения современных технологий и алгоритмов машинного обучения.

Для достижения поставленной цели решаются следующие **задачи**:

1. Аналитический обзор методов решения задач оптимального управления робототехническими системами и существующих подходов к их практической реализации.
2. Разработка принципа синтезированного оптимального управления, обеспечивающего получение практически реализуемого решения задачи оптимального управления в робототехнических системах.
3. Математическая формулировка задачи оптимального управления на основе принципа синтезированного оптимального управления.
4. Разработка двухэтапного подхода синтеза систем управления на основе принципа синтезированного оптимального управления.
5. Постановка задач управления на каждом этапе двухэтапного подхода и обзор существующих методов решения поставленных задач.

6. Формализованное обоснование применения алгоритмов машинного обучения управления при разработке систем управления робототехническими системами на основе принципа синтезированного оптимального управления.
7. Разработка методов машинного обучения управления на основе символьной регрессии для реализации этапа синтеза системы стабилизации в рамках принципа синтезированного оптимального управления.
8. Разработка алгоритмов оптимизации на основе современных эволюционных и популяционных подходов для решения задачи оптимального расположения точек равновесия в рамках принципа синтезированного оптимального управления.
9. Исследование свойства реализуемости решения задачи оптимального управления на основе применения принципа синтезированного оптимального управления.
10. Разработка программных комплексов для получения законов управления робототехническими системами на основе принципа синтезированного оптимального управления и проведение математического и имитационного моделирования движения робототехнических объектов.

Объектом исследования являются системы управления робототехническими устройствами.

Предметом исследования служат вычислительные методы для разработки оптимальных реализуемых систем управления робототехническими системами.

Методы исследования. В диссертационной работе использовались методы теории оптимального управления, математического анализа, методы численного моделирования, методы машинного обучения и вычислительные методы оптимизации.

Научная новизна.

1. Сформулирован новый принцип синтезированного оптимального управления. Согласно принципу синтезированного оптимального управления, решение задачи оптимального управления производится для объекта, стабилизированного относительно точки равновесия в пространстве состояний. Разработанный принцип синтезированного оптимального управления отвечает современным требованиям цифровой трансформации и автоматизирует процесс создания систем управления за счет внедрения универсальных технологий машинного обучения при использовании классических формулировок задач управления.
2. Разработан подход двухэтапной реализации принципа синтезированного оптимального управления в робототехнических системах. Для этой цели первоначально решается задача синтеза управления, чтобы обеспечить стабилизацию робототехнического объекта относительно точки в пространстве состояний, а затем решается задача оптимального управления. Оптимальное управление реализуется за счет оптимального изменения положения устойчивой точки равновесия. Представлены основные преимущества принципа синтезированного оптимального управления. Продемонстрирована универсальность предлагаемого подхода и его применимость к различным задачам оптимального управления робототехническими объектами.
3. Приведено обоснование применения принципа синтезированного управления для получения решения задачи оптимального управления, обладающего свойством реализуемости за счет обеспечения в каждый момент времени существования у объекта устойчивой точки равновесия.
4. Разработаны новые численные методы реализации этапов синтезированного оптимального управления средствами эволюционного машинного обучения. Разработка законов управления робототехническими объектами, согласно разработанному подходу, происходит автоматически. Для решения задачи синтеза управления и

обеспечения устойчивости объекта используется машинное обучение методом символьной регрессии. Для решения задачи глобальной оптимизации при определении оптимального положения точек равновесия используются специально отобранные эволюционные алгоритмы.

5. В рамках реализации принципа синтезированного оптимального управления разработаны новые вариационные методы машинного обучения на основе символьной регрессии для структурно-параметрического синтеза системы стабилизации робототехническими объектами, предложены уникальные типы малых вариаций и способы их кодирования. Разработаны программные комплексы их реализации.

Достоверность научных результатов, приведенных в диссертационной работе, подтверждается соответствием теоретических и экспериментальных результатов, проводимых как на математических моделях, так и на опытных образцах Роботоцентра ФИЦ ИУ РАН, а также экспертизой научных статей, опубликованных в ведущих научных российских и международных изданиях, апробацией и обсуждением результатов на международных и российских научных конференциях и семинарах.

Теоретическая и практическая значимость исследования. Предложенный новый принцип синтезированного оптимального управления позволяет разрабатывать законы управления для робототехнических систем на основе формализованных математических постановок с учетом критерия оптимальности и обладающие свойством реализуемости. Этап синтеза системы стабилизации является ключевой идеей подхода и обеспечивает достижение требуемых результатов в задачах с неопределенностями, которые неизбежно существуют в реальных системах. Вблизи точки равновесия все решения сжимаются, а значит выполняются введенные в работе дополнительные условия реализуемости оптимального управления. Применение алгоритмов машинного обучения позволяют автоматизировать процесс разработки законов управления, делают представленный подход универсальным и позволяют применять его к

различным нелинейным моделям объектов и функционалам любой сложности. а Предложенные подходы и численные методы, разрабатываемые в диссертации, имеют широкие возможности развития и исследования, как в рамках рассматриваемых задач управления робототехническими системами, так и в рамках других активно развивающихся направлений робототехники.

Результаты, представленные диссертации в главах 2 и 4 в части обоснования и исследования свойства реализуемости моделей систем управления, исследования разрабатываемых численных методов синтеза систем управления в классе реализуемых систем и проведения вычислительных экспериментов на моделях робототехнических систем были получены в рамках проекта Министерства науки и высшего образования Российской Федерации 075-15-2024-544 "Математические модели и численные методы как основа для разработки робототехнических комплексов, новых материалов и интеллектуальных технологий конструирования".

Разработанные в рамках диссертационного исследования методы были реализованы в виде программных модулей и используются в исследованиях, проводимых в Роботоцентре ФИЦ ИУ РАН по государственному заданию № 0063-2019-0010.

Результаты работы использованы в Инжиниринговом центре «Автоматика и робототехника МГТУ им. Н.Э. Баумана», в ФАУ ЦАГИ, в ООО «Научно-производственное объединение НаукаСофт», АО «ВПК «НПО Машиностроения», что подтверждается актами о внедрении, представленными в Приложении 3.

Соответствие диссертации паспорту научной специальности. В соответствии с формулой специальности 2.3.1 «Системный анализ, управление и обработка информации, статистика» (технические науки) диссертация посвящена разработке и исследованию принципа синтезированного оптимального управления на основе методов машинного обучения для робототехнических систем, что соответствует следующим пунктам паспорта специальности: п.2 «Формализация и постановка задач системного анализа,

оптимизации, управления, принятия решений, обработки информации и искусственного интеллекта», п.4 «Разработка методов и алгоритмов решения задач системного анализа, оптимизации, управления, принятия решений, обработки информации и искусственного интеллекта», п.7 «Методы и алгоритмы структурно-параметрического синтеза и идентификации сложных систем».

Основные положения, выносимые на защиту:

1) Принцип синтезированного оптимального управления, реализующий управление на основе обеспечения устойчивости объекта управления относительно точки равновесия в пространстве состояний и оптимального расположения устойчивой точки равновесия. Предложенный подход автоматизирует процесс создания систем управления за счет применения технологий машинного обучения.
2) Двухэтапный подход разработки системы управления на основе принципа синтезированного оптимального управления, который обеспечивает нахождение управления как функцию от состояния объекта, но в отличие от общего синтеза является более адаптивным к изменениям условий функционирования.
3) Обоснование обеспечения свойства реализуемости, согласно которому ошибка состояния объекта по модели не увеличивается во времени для систем управления, полученных на основе использования принципа синтезированного оптимального управления.
4) Численные методы реализации этапов принципа синтезированного оптимального управления на основе интеллектуальных алгоритмов машинного обучения. На первом этапе решается задача синтеза системы стабилизации объекта управления с целью обеспечения его устойчивости относительно точки равновесия в пространстве состояний. Для решения этой задачи используется машинное обучение управления методами символьной регрессии. На втором этапе решается задача оптимального размещения точек равновесия. Для решения этой

задачи используются эволюционные алгоритмы. Разработанные численные методы расчета синтезированного оптимального управления являются универсальными, не зависит от типа модели объекта управления и целевого функционала, что обеспечивает автоматизацию процесса построения реализуемой системы управления, причем задача оптимального размещения точек равновесия может решаться в режиме реального времени.

5) Разработанные новые вариационные методы машинного обучения на основе символьной регрессии и принципа малых вариаций для структурно-параметрического синтеза системы управления, позволяющие автоматизировать этап синтеза системы стабилизации. Методы имеют уникальные типы малых вариаций и способы их кодирования, а также специальный генетический алгоритм структурно-параметрического поиска функций, позволяющий реализовывать поиск одновременно и структуры функции управления, и ее параметров.
6) Программные комплексы, реализующие представленные в диссертационном исследовании методы машинного обучения для робототехнических объектов.

Апробация результатов.

Результаты диссертационного исследования докладывались и обсуждались на международных и всероссийских конференциях, семинарах и симпозиумах:

1. European Control Conference (ECC) July 15-17, 2015. Linz, Austria.
2. IFAC Conference on Modelling, Identification and Control of Nonlinear Systems MICNON 2015 Saint Petersburg, Russia, 24-26 June 2015.
3. IFAC Symposium on Robot Control, Salvador, BA, Brazil, August 26-28, 2015.
4. Вторая молодежная научная конференция «Задачи современной информатики», ФИЦ ИУ РАН, Москва, 2015.

5. Байкальская Всероссийская конференция с международным участием "Информационные и математические технологии в науке и управлении", 1-8 июля, 2016, Байкал, Иркутск, Россия.
6. International Conference on Control, Decision and Information Technologies: CoDIT 2024 (1 - 4 July 2024, Valletta, Malta), CoDIT 2022 (May 17 – 20, Istanbul, Turkey), CoDIT 2020 (June 29 - July 2, Prague, Czech Republic), CoDIT 2019 (April 23-26, Paris, France), CoDIT-2017 (April 5-7, 2017, Barcelona, Spain), CoDIT-2016 (6-8 April, Malta).
7. International Conference Optimization and Applications: OPTIMA-2020 (Moscow, Russia, September 28 – October 2), OPTIMA-2017 (Petrovac, Montenegro, October 2-7).
8. Intelligent Systems Conference (IntelliSys), 3-4 September 2020, Virtual Event.
9. International Conference “Intelligent Systems”: INTELS-2022 (December 14–16, Moscow), INTELS-2020 (December 14–16, 2020, Virtual), INTELS-2018 (October 22-24, Saint Petersburg), INTELS-2016 (5–7 October, Moscow), INTELS-2014 (30 June - 4 July, Moscow).
10. IEEE Conference on Industrial Electronics and Applications: ICIEA-2020 (9-13 Nov. 2020, Kristiansand, Norway), ICIEA-2019 (19-21 June 2019, Xi’an, China), ICIEA-2016 (5 - 7 June 2016, Hefei, China).
11. Всероссийская мультиконференция по проблемам управления: МКПУ-2023 (11 сентября - 15 сентября, Волгоград, Россия), МКПУ-2021 (27 сентября - 02 октября, с. Дивноморское, Геленджик, Россия), МКПУ-2019 (23 - 28 сентября, с. Дивноморское, Геленджик, Россия), МКПУ-2017 (11 - 16 сентября, с. Дивноморское, Геленджик, Россия), МКПУ-2015 (28 сентября – 03 октября, с. Дивноморское, Геленджик, Россия).
12. Международный симпозиум «Надежность и качество»: 27 мая - 01 июня 2024 г., 29 мая – 2 июня 2023 г., 23–31 мая 2022 г., 23 мая – 31 мая 2016 г., г. Пенза, Россия.
13. Международная научно-практическая конференция «Фундаментально-прикладные проблемы безопасности, живучести, надёжности,

устойчивости и эффективности систем», 11 сентября 2020 г., 3-5 июня 2019 г., 4-7 июня 2018 г., 1-4 февраля 2017 г., 2-4 июня 2015, 20-22 ноября 2014 г., Елец, Россия.

14. Всероссийская конференция с международным участием «Прикладные проблемы системной безопасности», 21-22 сентября 2023, Елец, Россия.
15. Завалишинские чтения 2021: XVI международная конференция по электромеханике и робототехнике, С.-Петербург, Россия, 14-17 апреля.
16. Международная научно-техническая конференция "ЭКСТРЕМАЛЬНАЯ РОБОТОТЕХНИКА", 29-30 сентября 2022, Санкт-Петербург, Россия.
17. XIV Всероссийское совещание по проблемам управления ВСПУ-2024, Россия, Москва, ИПУ РАН, 17-20 июня 2024.
18. The Genetic and Evolutionary Computation Conference (GECCO-2024), Melbourne, Australia, July 14 - 18, 2024.

Публикации автора по теме диссертации. Основные результаты диссертационного и опубликованы в 90 научных публикациях, из них 1 монография, 12 публикаций в изданиях, включенных в перечень рецензируемых журналов, рекомендованных ВАК, включая 10 публикаций в изданиях, отнесенных к категориям К-1 или К-2 из Перечня ВАК, 34 – в научных изданиях, индексируемых Web of Science и Scopus, включая 7 статей в изданиях Q1 и Q2 квартиля, 43 – в трудах конференций и изданиях, включенных в базу РИНЦ, получено 15 свидетельств на программы для ЭВМ.

Личный вклад. Все положения, выносимые на защиту, изложенные в диссертации, принадлежат лично автору. В совместных работах автор принимал непосредственное участие в формировании направления исследования, теоретических обоснований, в программной реализации, в выборе экспериментальных задач и проведении вычислительных экспериментов.

Структура и объем диссертации. Диссертация состоит из введения, четырех глав, заключения, библиографического списка и приложений. Содержание работы изложено на 316 страницах, включает 5 таблиц, 70 рисунков и 3 приложения.

Глава 1. Проблемы автоматизации разработки законов управления робототехническими системами

В современном мире быстрых и эффективных компьютеров каждый день появляются все новые задачи, решение которых можно доверить машине вместо использования аналитических расчетов и ручного труда. Такие машинные решения универсальны и не привязаны к конкретным особенностям задачи, исключают появление ошибок из-за человеческого фактора. Нужно только формализовать условия и требования и далее доверить поиск решения машине, которая с помощью современных универсальных численных методов сможет получить решение, удовлетворяющее сформулированным требованиям.

Сегодня пришло время включить в круг таких «машинно реализуемых» задач и задачу разработки систем управления для роботов. Число и разнообразие роботов растет огромными темпами. Так собираемся ли мы программировать каждого робота вручную?

На современном этапе, начиная с 80-х годов XX века, роль устройства управления играет компьютер. Таким образом, цифровую автоматизацию, или цифровое автоматическое управление, можно определить как технологию, использующую запрограммированные команды, воздействующие на какой-либо объект или процесс, и обратную связь, с помощью которой определяется, правильно ли выполняются эти команды [2]. А разработка системы управления теперь заключается в программировании устройства управления.

На рис. 1.1 приведена схема цифровой системы управления, в которой роль управляющего устройства выполняет компьютер.

Рис. 1.1. Цифровая система управления

Обратим внимание, что серьезным фактором при реализации систем управления является получение достаточно надежных измерений всех существенных для процесса управления переменных состояния объекта. Это важное направление современных исследований в области разработки автоматических систем управления и робототехники, которому уделяется сейчас также большое внимание. В настоящее время существует множество видов навигации автономных беспилотных робототехнических систем. Спутниковая навигация, безусловно, является наиболее распространенным способом. Однако этот способ имеет свои ограничения применимости. Например, в помещении спутниковый сигнал обычно не доходит до устройств через бетонные и металлические конструкции. В этом случае можно использовать такие инструменты навигации, как триангуляция [3], навигация по различным маркерам, например QR-кодам [4, 5], штрих-коды [6], маркеры ArUco для коррекции положения [7 - 9] и SLAM-навигация [10, 11], а также комбинации вышеперечисленных методов. В неизведанных пространствах наиболее перспективной из упомянутых технологий является метод навигации SLAM [12]. Этот метод можно использовать в неподготовленной и неизвестной роботу комнате для создания карты и последующего ее использования. В настоящее время эта концепция широко используется в области робототехники для решения задач автономного движения [13 - 15]. По данным SLAM, роботу необходимо знать свое местоположение в каждый момент времени, а также постепенно сканировать окружающее пространство с помощью датчиков,

составляя таким образом карту местности. Карта строится постепенно по мере исследования роботом новых территорий. Основным источником информации о местоположении робота является одометрия, полученная тем или иным способом (энкодеры, компьютерное зрение и другие инерциальные средства навигации и их комбинация). Но любая сенсорная система имеет свои слабые места, например, камеры могут выйти из строя при изменении освещения или может произойти пробуксовка колес, что приведет к ошибке одометрии. В связи с этим возникает необходимость каким-то образом поддерживать навигационную систему, поэтому исследования этой области также сохраняют свою актуальность и востребованность [16 - 20]. Однако, данная область является отдельным направлением исследования. В рамках диссертационной работы предполагается, что на систему управления приходит достоверная оценка вектора состояния объекта управления *с блока измерений*. И основным направлением диссертационного исследования является разработка *блока системы управления* (см. рис. 1.1).

С появлением высокопроизводительных и относительно недорогих компьютеров изменились сами подходы к разработке систем управления. Ранее разработка системы управления заключалась в первоначальном формировании конфигурации системы (задании определенной структуры) и дальнейшем расчете и настройке оптимальных параметров, обеспечивающих желаемые показатели качества. Этап настройки конфигурации системы управления для конкретного объекта управления был сродни искусству, в процессе которого колоссальную роль играет опыт и интуиция инженера-разработчика. Такой подход был оправдан ранее, когда системы автоматического управления применялись в основном для очень дорогостоящих объектов, в частности, в ракетах или в автопилотах самолетов и подводных лодок. Сейчас же появились роботы, причем количество этих роботов с учетом аддитивных технологий катастрофически растет с каждым годом, а использование технических методов создания систем автоматического управления для них является основным препятствием для их быстрой разработки и широкого внедрения.

Написание вручную программы системы управления роботами становится все более сложной задачей. Так, например, если робот должен выполнять достаточно небольшой объем действий, скажем, перемещаться из одной точки в другую и объезжать какие-то препятствия, то программа его системы управления содержит несколько сотен строк. В более сложных задачах управления программы, которые должны управлять роботами, могут включать несколько десятков или сотен тысяч строк. Эти программы будут расширяться по мере усложнения задач или структуры роботов.

Например, сколько операторов будет содержать программа управления роботом, имитирующая действия мухи? Муха управляет сложным движением крыльев, что позволяет ей неподвижно висеть в воздухе, она может двигаться по вертикальной поверхности. Далее муха видит опасности и совершает сложные движения, чтобы не быть пойманной. При этом, как и обычное животное, она ищет пищу и возможность размножения. По простой, самой оптимистической оценке, система управления таким объектом должна содержать более миллиона программных операторов. Вероятно, такую программу могла бы написать большая команда программистов. Но и здесь, и при создании еще более сложных систем управления, очевидна необходимость автоматизации синтеза системы управления.

По факту же на практике в робототехнике большинство современных систем управления роботами программируются вручную, и инженеры даже не ставят общих задач, потому что нет общих способов их решения. Разработчик, исходя из своего опыта, задает структуру системы управления, определяет каналы управления, типы регуляторов, а затем настраивает параметры данной системы так, чтобы они соответствовали определенным требованиям [21]. Однако задачу разработки системы управления тоже можно и нужно автоматизировать, определяя не только параметры, но и структуру системы управления оптимально и автоматически.

Возникает вопрос: можем ли мы доверить машине самой, без помощи инженера, задающего структуру системы управления, автоматически

определить эту структуру, исследуя данные или опираясь на заданные оценки или желаемые характеристики системы?

Этот вопрос открывает дверь к новой парадигме разработки систем управления – автоматизированной разработки управления с помощью современных методов машинного обучения управлению.

Для этого необходимо формализовать условия и требования и далее доверить поиск решения машине. Фундаментом для формальной постановки задач управления является теория оптимального управления, в которой основная задача направлена на поиск закона управления для заданной системы, обеспечивающего выполнение цели управления с оптимальным значением заданного критерия качества.

Различные задачи для роботов, как и любых других объектов управления, можно и нужно формулировать как задачи оптимального управления, например, для нахождения оптимального пути в текущих реальных условиях, задачи предотвращения столкновений со статическими и динамическими препятствиями, задачи взаимодействия с другими объектами управления, задачи точного достижения заданных терминальных условий и т.д.

1.1. Задачи оптимального управления в робототехнических системах

Рассмотрим наиболее востребованные постановки задач управления для робототехнических систем. Описание задач представим в терминах, удобных для программирования и решения этих задач численными методами.

Классическая задача оптимального управления возникает в тех случаях, когда необходимо перевести одного или нескольких роботов из одного заданного состояния в другое.

Задана математическая модель объекта управления в виде системы обыкновенных дифференциальных уравнений, записанных в форме Коши

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}), \tag{1.1}$$

где $\mathbf{x}$ – вектор состояния объекта управления, $\mathbf{x} \in \mathbb{R}^{\boldsymbol{n}}$, $\mathbf{u}$ – вектор управления, все компоненты вектора управления, как правило, ограничены снизу и сверху

$$\mathbf{u}^- \le \mathbf{u} \le \mathbf{u}^+, \tag{1.2}$$

$\mathbf{u}^- = [u_1^- \dots u_m^-]^{\boldsymbol{T}}$, $\mathbf{u}^+ = [u_1^+ \dots u_m^+]^{\boldsymbol{T}}$ – значения нижних и верхних ограничений, m – размерность вектора управления. В общем виде ограничения (1.2) записывают как принадлежность компактному множеству U, которое представляет собой ограниченную область в вещественном пространстве $\mathbb{R}^{\boldsymbol{m}}$, причем границы этой области также принадлежат компактному множеству

$$\mathbf{u} \in \mathrm{U} \subseteq \mathbb{R}^{\boldsymbol{m}}. \tag{1.3}$$

Для системы дифференциальных уравнений (1.1) или объекта управления задано начальное состояние

$$\mathbf{x}(0) = \mathbf{x}^0 = [x_1^0 \dots x_n^0]^T. \tag{1.4}$$

Для системы дифференциальных уравнений (1.1) задано терминальное состояние

$$\mathbf{x}(t_f) = \mathbf{x}^f = [x_1^f \dots x_n^f]^T, \tag{1.5}$$

где t_f – время достижения терминального состояния.

В большинстве задач для робототехнических систем терминальное время t_f не задано, но ограничено

$$t_f \le t^+, \tag{1.6}$$

где t^+ - заданное предельное время достижения терминального состояния.

В практических задачах момент достижения терминального состояния определяем с некоторой погрешностью, величина, которой задана. С учетом заданной погрешности определим соотношения для вычисления терминального времени

$$t_f = \begin{cases} t, \text{если } t < t^+ \text{ и } \left\|\mathbf{x}(t, \mathbf{x}^0) - \mathbf{x}^{\boldsymbol{f}}\right\| \le \varepsilon_1 \\ t^+ - \text{иначе} \end{cases}, \tag{1.7}$$

где $\mathbf{x}(t, \mathbf{x}^0)$ – частное решение дифференциального уравнения (1.1) из начального состояния $\mathbf{x}^0$, (1.4), ε_1 – заданная допустимая погрешность попадания в терминальное состояние,

$$\left\|\mathbf{x}(t, \mathbf{x}^0) - \mathbf{x}^{\boldsymbol{f}}\right\| = \sqrt{\sum_{i=1}^{n} (x_i(t, \mathbf{x}^0) - x_i^f)^2}. \tag{1.8}$$

Задан критерий качества в интегральной форме

$$J_0 = \int_0^{t_f} f_0(\mathbf{x}(t, \mathbf{x}^0), \mathbf{u})dt \to \min_{\mathbf{u}\in \mathrm{U}}. \quad (1.9)$$

Классическая задача оптимального управления для группы одинаковых роботов имеет приблизительно такое же описание с дополнительным условием отсутствия столкновения между роботами. В результате получаем такую же задачу оптимального управления, но большей размерности. Приведем описание задачи оптимального управления для группы роботов.

Задана математическая модель всех объектов управления, группы роботов

$$\dot{\mathbf{x}}^j = \mathbf{f}\left(\mathbf{x}^j, \mathbf{u}^j\right), j = 1, \dots, N, \quad (1.10)$$

где $\mathbf{x}^j$ – вектор состояний робота j, $\mathbf{u}^j$ – вектор управлений робота j, $j = 1, \dots, N$, N – количество роботов в группе.

Для значения вектора управлений роботами заданы ограничения

$$\mathbf{u}^j \in \mathrm{U} \subseteq \mathbb{R}^{\boldsymbol{m}}, j = 1, \dots, N, \quad (1.11)$$

Для каждого робота задано начальное состояние

$$\mathbf{x}^j(0) = \mathbf{x}^{j,0} = [x_1^{j,0} \dots x_n^{j,0}]^T, j = 1, \dots, N, \quad (1.13)$$

Для каждого робота задано терминальное состояние

$$\mathbf{x}^j\left(t_{f,j}\right) = \mathbf{x}^{j,f} = [x_1^{j,f} \dots x_n^{j,f}]^T, j = 1, \dots, N, \quad (1.14)$$

где $t_{f,j}$ – терминальное время или время достижения терминального состояния $\mathbf{x}^{j,f}$ роботом $j, j = 1, \dots, N$. Терминальное время $t_{f,j}$ определяем из соотношения аналогичного (1.7)

$$t_{f,j} = \begin{cases} t, \text{если } t < t^+ \text{ и } \left\|\mathbf{x}^j\left(t, \mathbf{x}^{j,0}\right) - \mathbf{x}^{j,f}\right\| \le \varepsilon_1 \\ t^+ - \text{иначе} \end{cases}, j = 1, \dots, N. \quad (1.15)$$

Заданы ***динамические фазовые ограничения***, которые описывают условия отсутствия столкновений между роботами

$$\chi(\mathbf{x}^{j_1}, \mathbf{x}^{j_2}) \le 0, j_1 = 1, \dots, N-1, j_2 = j_1 + 1, \dots, N, \quad (1.16)$$

где $\chi(\mathbf{x}^{j_1}, \mathbf{x}^{j_2})$ – функция, определяющая условие, чтобы расстояние между роботами в геометрическом пространстве не превышало заданную величину, поэтому при вычислении расстояния между роботами используются не все

компоненты векторов состояний, а только те, которые определяют координаты центров масс роботов в геометрическом пространстве

$$\chi\left(\mathbf{x}^{j_1}, \mathbf{x}^{j_2}\right) = r_0 - \sqrt{\sum_{i=1}^{k} (x_i^{j_1}(t, \mathbf{x}^{j_1,0}) - x_i^{j_2}(t, \mathbf{x}^{j_2,0}))^2}, \quad (1.17)$$

где $j_1, j_2 \in \{1, \dots, N\}$, $j_1 \neq j_2$, r_0 - заданная положительная величина, определяющая минимальное расстояние между центрами масс роботов, в рассматриваемом геометрическом пространстве размерности $k \leq 3$.

Задан критерий качества в интегральной форме

$$J_1 = \int_0^{t_f} f_0(\mathbf{x}^1(t, \mathbf{x}^{1,0}), \dots, \mathbf{x}^N(t, \mathbf{x}^{N,0}), \mathbf{u}^1, \dots, \mathbf{u}^N)dt \to \min_{\mathbf{u}^1, \dots, \mathbf{u}^N \in \mathrm{U}}, \quad (1.18)$$

где $t_f = \max\{t_{f,1}, \dots, t_{f,N}\}$. При вычислении считаем, что, если робот j за время $t_{f,j}$ достиг терминального состояния, то он остается в этом состоянии до тех пор, пока остальные роботы не достигнут своих терминальных состояний

$$\mathbf{x}\left(t, \mathbf{x}^{j,0}\right) = \mathbf{x}^{j,f}, t_{f,j} \leq t \leq t_f. \quad (1.19)$$

В некоторых задачах оптимального управления роботами не задается терминальное состояние, например в задаче мониторинга территории, когда робот просматривает территорию для поиска определенных предметов или для картографирования. В этом случае терминальное время определяется из дополнительных условий. Для одного робота условия окончания процесса управления имеют следующий вид:

$$t_f = \begin{cases} t, \text{если } t < t^+ \text{ и } \sigma\left(\mathbf{x}(t, \mathbf{x}^0)\right) = 0 \\ t^+ - \text{иначе} \end{cases}. \quad (1.20)$$

Для робота j из группы роботов условия остановки процесса управления имеют аналогичный вид

$$t_{f,j} = \begin{cases} t, \text{если } t < t^+ \text{ и } \sigma\left(\mathbf{x}^j\left(t, \mathbf{x}^{j,0}\right)\right) = 0 \\ t^+ - \text{иначе} \end{cases}, j \in \{1, \dots, N\}. \quad (1.21)$$

В формулах (1.20), (1.21) функция $\sigma(\mathbf{x}(t))$ определяет условия завершения управляемого процесса в момент $t < t^+$, например условия полного просмотра области.

В задачах оптимального управления роботами, как правило, учитывают ***фазовые ограничения***, которые описывают в пространстве, где функционируют роботы, некоторые области, куда робот не должен попадать. Например, для квадрокоптера необходимо задать верхнее и нижнее ограничения высоты полета. В других случаях это могут быть препятствия, попадание в которые не желательно.

В общем случае условия удовлетворения фазовых ограничений определим в виде функции:

$$\varphi_i(\mathbf{x}(t,\mathbf{x}^0)) \le 0,\ i = 1,\dots,M,\ 0 \le t \le t_f, \qquad (1.22)$$

где M - количество фазовых ограничений.

Для группы роботов фазовые ограничения необходимо проверять для каждого робота из группы

$$\varphi_i\left(\mathbf{x}^j(t,\mathbf{x}^{j,0})\right) \le 0,\ i = 1,\dots,M,\ 0 \le t \le t_{f,j},\ j = 1,\dots,N. \qquad (1.23)$$

Пусть некоторая компонента x_k, $k \in \{1,\dots,n\}$, вектора состояния робота в процессе функционирования должна иметь ограниченные снизу и сверху значения, $x_k^- \le x_k \le x_k^+$. Тогда функции таких линейных фазовых ограничений (1.22) имеют следующий вид

$$\varphi_1(\mathbf{x}(t,\mathbf{x}^0)) = x_k - x_k^+ \le 0,\ \varphi_2(\mathbf{x}(t,\mathbf{x}^0)) = x_k^- - x_k \le 0. \qquad (1.24)$$

Если фазовые ограничения описывают ограниченную область в геометрическом пространстве и эта область может быть помещена в сферу соответствующей размерности, то для описания таких фазовых ограничений целесообразно использовать следующие функции:

$$\varphi_3(\mathbf{x}(t,\mathbf{x}^0)) = r_3 - \sqrt{\sum_{i=1}^{k}(x_i(t,\mathbf{x}^0) - y_{3,i})^2} \le 0, \qquad (1.25)$$

где r_3 – радиус сферического препятствия в k-мерном геометрическом пространстве, $y_{3,i}$, $i = 1,\dots,k$ – координаты центра сферического препятствия в k-мерном геометрическом пространстве, $x_i(t,\mathbf{x}^0)$, $i = 1,\dots,k$, координаты центра масс робота, соответственно для препятствия в пространстве $k = 3$, для препятствия на плоскости $k = 2$.

Другим типом фазовых ограничений в задачах управления роботами являются зоны обязательного прохождения. В отличие от препятствий здесь наоборот требуется прохождение объекта управления через определенные области пространства. Как правило эти области имеют небольшой размер и в процессе управления робот должен как минимум один раз пройти через эту область. Для установления факта прохождения через область определяем в процессе управления минимальное расстояние до заданной области. Если это минимальное расстояние не превышает заданную величину, то считаем, что область пройдена. Области обязательного прохождения определяем координатами точки центра этой области и заданной величиной минимального расстояния до центра области. Для определения условия выполнения таких фазовых ограничений вычисляем расстояние до области в начальный момент времени

$$d^- = \sqrt{\sum_{i=1}^{k}(z_i - x_i^0)^2}, \tag{1.26}$$

где z_i – координаты центра заданной области в пространстве размерности k.

Далее вычисляем это расстояние в каждый последующий момент времени

$$d(t) = \sqrt{\sum_{i=1}^{k}(z_i - x_i(t, \mathbf{x}^0))^2} \tag{1.27}$$

и сравниваем его с текущим наименьшим расстоянием. Если $d(t) < d^-$, $d^- = d(t)$.

В качестве ***основных критериев*** в задачах оптимального управления робототехническими системами рассматриваем:

- критерий быстродействия

$$J = t_f = \int_0^{t_f} 1dt \to \min, \tag{1.28}$$

- критерий минимальной длины траектории

$$J = \int_0^{t_f} \sqrt{\sum_{i=1}^{k} \dot{x}_i^2}\, dt \to \min, \tag{1.29}$$

где $\mathbb{R}^k$ – подпространство, в котором измеряется длина пути траектории

- критерий минимальной ошибки движения по заданной траектории во времени

$$J = \int_0^{t_f} \sqrt{\sum_{i=1}^{k} (x_i^*(t) - x_i(t, \mathbf{x}^0))^2}\, dt \to \min, \tag{1.30}$$

или

$$J = \max_{t \in \{0, t_f\}} \sqrt{\sum_{i=1}^{k} (x_i^*(t) - x_i(t, \mathbf{x}^0))^2} \to \min, \tag{1.31}$$

- критерий минимальной ошибки движения по заданной траектории, если траектория задана в виде множества точек в подпространстве $\mathbb{R}^k$

$$J = \sum_{j=1}^{S} \min_{t \in \{0, t_f\}} \sqrt{\sum_{i=1}^{k} (x_i^{*,j} - x_i(t, \mathbf{x}^0))^2} \to \min, \tag{1.32}$$

где S – количество точек на траектории в подпространстве $\mathbb{R}^k$,

- критерий минимального расхода управления (энергии, топлива и др.):

$$J = \int_0^{t_f} \mathbf{u}^{\boldsymbol{T}} \mathbf{C} \mathbf{u} dt \to \min, \tag{1.33}$$

где $\mathbf{C}$ – диагональная матрица весов компонент управления.

- квадратичный критерий по состоянию:

$$J = \int_0^{t_f} \mathbf{x}^{\boldsymbol{T}} \mathbf{D} \mathbf{x} dt \to \min, \tag{1.34}$$

где $\mathbf{D}$ – диагональная матрица весов компонент состояния.

Все критерии (1.28) – (1.34) могут быть частью составного критерия оптимальности.

При решении задачи оптимального управления на основе численного прямого подхода, в котором задачу оптимального управления преобразуем к задаче нелинейного программирования, к основному критерию качества добавляем с коэффициентами штрафа точность попадания в терминальное состояние (1.8) и условия выполнения фазовых ограничений (1.17), (1.22), (1.23).

При включении в критерий качества выполнения условий фазовых ограничений используем функцию Хэвисайда

$$\vartheta(\alpha) = \begin{cases} 1, \text{если } \alpha > 0 \\ 0 - \text{иначе} \end{cases}, \tag{1.35}$$

$$J = \int_0^{t_f} f_0(\mathbf{x}(t, \mathbf{x}^0), \mathbf{u})dt + p_1 \sum_{i=1}^{M} \int_0^{t_f} \vartheta(\varphi_i(\mathbf{x}(t, \mathbf{x}^0)))dt \to \min_{\mathbf{u} \in \mathrm{U}}. \quad (1.36)$$

Значение второго интеграла равно длительности времени нарушения фазовых ограничений.

Итак, в данном подразделе представлены задачи в робототехнических системах, сформулированные как задачи оптимального управления. Приведены в общем виде классические постановки задач оптимального управления для роботов и группы роботов. Приведены функции для описания условий, определяющих фазовые ограничения. Приведены критерии качества управления и правила включения фазовых ограничений в критерий качества.

1.2. Анализ подходов к решению задачи оптимального управления

Задача оптимального управления всегда должна и может быть поставлена, когда необходимо найти управление для динамического объекта, который должен выполнять некоторые действия с заданным критерием качества.

На сегодняшний день теория оптимального управления сложилась в мощную самостоятельную математическую дисциплину, которая располагает целым набором средств и методов решения задач оптимального управления для разных классов объектов.

В исходной классической постановке задача оптимального управления, представленная Л.С. Понтрягиным [22, 23], имеет следующий вид. Объект управления описывается системой обыкновенных дифференциальных уравнений, в правой части которой стоит неизвестный вектор управления. Также заданы начальные и конечные условия и интегральный функционал качества.

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}), \ \ \mathbf{x} \in \mathbb{R}^n, \\ \mathbf{x}(0) = \mathbf{x}^0, \\ \mathbf{x}(T) = \mathbf{x}^f, \\ J = \int_0^T f_0(\mathbf{x}, \mathbf{u})\, dt \to \min, \\ \mathbf{u} \in \mathrm{U} \subseteq \mathbb{R}^m, \end{cases} \quad (1.37)$$

где $\mathbf{x}$ - вектор, описывающий состояние объекта управления, $\mathbf{u}$ - управление, U - компактное множество, T - терминальное время, может быть не задано, но ограничено $T \leq t^+$, t^+ - заданная величина границы по времени.

В этой задаче (1.37) необходимо найти оптимальную функцию управления:

$$\mathbf{u} = \mathbf{v}(t) \in \mathrm{U}, \tag{1.38}$$

такую, что если ее подставить в правые части системы дифференциальных уравнений модели, то мы получим нестационарную систему дифференциальных уравнений с известной функцией времени в правой части

$$\dot{\mathbf{x}} = \mathbf{f}\big(\mathbf{x}, \mathbf{v}(t)\big), \tag{1.39}$$

и частное решение $\mathbf{x}(t, \mathbf{x}^0)$ этой системы (1.39) из заданного начального состояния $\mathbf{x}(0) = \mathbf{x}^0$ попадет в терминальное состояние $\mathbf{x}(T, \mathbf{x}^0) = \mathbf{x}^f$, $T \leq t^+$ с оптимальным значением критерия качества

$$J = \int_0^T f_0\left(\mathbf{x}(t, \mathbf{x}^0), \mathbf{v}(t)\right) dt = \min_{\mathbf{u} \in \mathrm{U}}.$$

Задача оптимального управления (1.37) - (1.8) относится к задачам бесконечномерной оптимизации, поскольку необходимо найти управление как вектор-функцию времени.

Исторически, эта задача принадлежала к классу задач вариационного исчисления. Различные задачи оптимального управления динамическими процессами можно с успехом решать с помощью классического вариационного исчисления в случаях открытого множества допустимых управлений U и предположений гладкости искомых управлений [24 - 26]. Об истории расширения классического вариационного исчисления на задачи оптимального управления можно прочитать в [27 - 30]. Целое созвездие знаменитых математиков как на Западе, так и в Советском Союзе, внесли значительный вклад в эту теорию [31 - 34], среди которых хотелось бы особо отметить основополагающие работы Л. Эйлера, Ж.-Л. Лагранжа и не ограничивая при этом важности работ других ученых: Д. Гильберта, А.М. Лежандра, К. Якоби, К. Вейерштрасса, Б. Римана, А. Лебега, Л. Тонелли, Н.Н. Боголюбова, С.Н. Бернштейна, М.А. Лаврентьева, А.Г. Сигалова, С.Ф. Морозова, В. Ритца,

Л. Янга, Дж. Варги, Э. Макшейна, Р.В. Гамкрелидзе, А.Д. Иоффе, В.Ф. Кротова, А.М. Размадзе, В.М. Тихомирова и др.

Однако, в общем случае, ограничения на значения управления и возможность у функции управления иметь разрывы первого рода как по величине, так и по производным ограничивают возможность применения методов вариационного исчисления к задаче оптимального управления. А так как ограничения имеют исключительно большое практическое значение, то важно найти подходы, которые позволят их учитывать.

На сегодняшний день разработаны различные как прямые, так и непрямые подходы к решению исходной задачи оптимального управления (1.37) –(1.38).

Прямой подход заключается в сведении задачи оптимального управления к задаче нелинейного программирования [35 - 37] через дискретизацию функции управления и состояния на временной сетке.

$$\mathbf{u} = \mathbf{v}(t, \mathbf{q}), \tag{1.39}$$

где $\mathbf{q}$ -вектор параметров, $\mathbf{q} \in \mathbb{R}^p$.

Для реализации прямого подхода можно применять любую кусочно-полиномиальную аппроксимацию, в частности можно использовать кусочно-линейную аппроксимацию функций управления.

Приведем соотношения для кусочно-линейной аппроксимации функций управления для решения задачи оптимального управления на основе прямого подхода. Для этой цели задаем интервал времени Δt и разбиваем ось времени на интервалы. Всего получаем

$$K = \left\lfloor \frac{t^+}{\Delta t} \right\rfloor \tag{1.40}$$

интервалов. Все интервалы имеют длительность Δt кроме последнего. Последний интервал имеет длительность

$$\Delta_K t = t^+ - (K-1)\Delta t. \tag{1.41}$$

Задаем на границах интервалов значения постоянных параметров для каждой компоненты вектора управлений. В результате получаем $K+1$ группу

из m параметров. Всего получаем $p = m(K+1)$ компонентов вектора искомых параметров

$$\mathbf{q} = [q_1 \dots q_p]^T. \quad (1.42)$$

Каждая $i + (j-1)m$-я компонента вектора параметров, $i = 1, \dots, m$, $j = 1, \dots, K$, соответствует компоненте u_i, $i \in \{1, m\}$, вектора управлений.

Для значений вектора параметров задаем ограничения

$$q_k^- \le q_k \le q_k^+, k = 1, \dots, p, \quad (1.43)$$

где q_k^- и q_k^+, соответственно нижняя и верхняя граница значений параметра q_k, $k = 1, \dots, p$.

Ограничения на вектор параметров должны превосходить ограничения на соответствующую компоненту вектора управлений

$$q_{i+(j-1)m}^- < u_i^-, i = 1, \dots, m, j = 1, \dots, K, \quad (1.44)$$

$$u_i^+ < q_{i+(j-1)m}^+, i = 1, \dots, m, j = 1, \dots, K, \quad (1.45)$$

Это сделано для того, чтобы обеспечить высокую скорость изменения управления на интервале, т.е. обеспечить большую величину так называемой липшицевости функции, и приблизить ее реализацию к функции с разрывами первого рода.

На каждом интервале значения функции управления определяем из соотношения

$$u_i(t) = \begin{cases} u_i^-, \text{если } \tilde{u}_i(t) \le u_i^- \\ u_i^+, \text{если } u_i^+ \le \tilde{u}_i(t), i = 1, \dots, m, \\ \tilde{u}_i(t), \text{иначе} \end{cases} \quad (1.46)$$

где

$$\tilde{u}_i(t) = \frac{q_{i+jm} - q_{i+(j-1)m}}{\Delta t}(t - j\Delta t) + q_{i+jm}, \quad (1.47)$$

$(j-1)\Delta t \le t < j\Delta t, i = 1, \dots, m, j = 1, \dots, K$.

Задача оптимального управления при прямом подходе с включенными в критерий качества фазовыми ограничениями становится задачей нелинейного программирования с несколькими локальными минимумами на пространстве $p = m(K+1)$ параметров, т.е. относится к классу задач глобальной

оптимизации. Таким образом обеспечивается переход от задачи оптимизации в бесконечномерном пространстве к задаче оптимизации в конечномерном пространстве [38, 39]. В результате мы получаем задачу нелинейного программирования большой размерности. Для решения такой задачи необходимо применять численные глобальной оптимизации.

Большинство прикладных задач оптимального управления имеют неунимодальный функционал, особенно задачи с фазовыми ограничениями, которые часто встречаются при управлении роботами, особенно в задачах управления группами роботов, где каждый робот является динамическим фазовым ограничением для других роботов. Пространство возможных решений в таком случае становится невыпуклым. Универсальных алгоритмов решения таких задач нелинейного программирования не существует, метод решения выбирается в каждом конкретном случае в зависимости от вида целевой функции, сложности ее вычисления, имеющихся ограничений, необходимой точности решения, мощности имеющейся вычислительной машины и т.д. Ведущее место среди прямых методов решения экстремальных задач занимают градиентные методы [40, 41], однако в случаях неунимодальности и невыпусклости функционала они имеют известные ограничения, связанные с нахождением локального экстремума [42]. При решении реальных задач даже приблизительная оценка глобального оптимума оказывается неизвестной. Это обстоятельство определяет популярность современных алгоритмов глобальной оптимизации [43 - 52], как детерминированных, развиваемых в работах Ю.Г. Евтушенко, Р.Г. Стронгина, М.А. Посыпкина, И.Х. Сигала, Х. Туя и других российских и зарубежных ученых [42 - 47], так и в особенности эвристических методов, основанных на популяционном и эволюционном поиске, представленных в работах Дж. Холланда, Л. Фогеля, Д. Гольдберга, Е.С. Семенкина, А.П. Карпенко и других исследователей [48 - 50], с точки зрения параметрического поиска функции управления в задачах с фазовыми ограничениями [51, 52]. Но в этом случае за счет параметризации управления можно получить только приближенное решение.

Основной классический подход решения задачи оптимального управления часто называют непрямым подходом. Он основан на применении принципа максимума Понтрягина. В настоящее время принцип максимума Понтрягина является наиболее значимым результатом в области оптимального управления. Он был сформулирован в 50-е годы XX века [22]. Принцип максимума требует относительно слабых предположений о дифференцируемости, хорошо подходит для задач с ограничениями, а также позволяет учитывать «гладкие» множества конечных состояний [53].

Применение принципа максимума к задаче оптимального управления позволяет рассматривать задачу оптимального управления как задачу оптимизации в конечномерном пространстве. Согласно принципу максимума в задачу вводятся сопряженные переменные $\boldsymbol{\psi} = [\psi_1 \dots \psi_n]^T$, которые в данном случае играют роль множителей Лагранжа в задаче условной минимизации. Сопряженные переменные необходимы для того, чтобы при минимизации заданного функционала учесть связи, описываемые математической моделью объекта управления. В отличие от множителей Лагранжа сопряженные переменные меняются со временем, и для их определения необходимо решить систему дифференциальных уравнений. Далее строится гамильтониан H, включающий в себя сопряженные переменные и правые части системы дифференциальных уравнений модели объекта управления.

$$H(\mathbf{x}, \boldsymbol{\psi}, \mathbf{u}) = -f_0(\mathbf{x}, \mathbf{u}) + \boldsymbol{\psi}^T \mathbf{f}(\mathbf{x}, \mathbf{u}), \tag{1.48}$$

где $\boldsymbol{\psi}$ – вектор сопряженных переменных, $\boldsymbol{\psi} = [\psi_1 \dots \psi_n]^T$. Вектор сопряженных переменных определяется из системы дифференциальных уравнений

$$\dot{\boldsymbol{\psi}} = -\frac{\partial H(\mathbf{x}, \boldsymbol{\psi}, \mathbf{u})}{\partial \mathbf{x}}. \tag{1.49}$$

Согласно принципу максимума, необходимое условие оптимального управления - максимизировать значение гамильтониана в каждый момент времени:

$$\mathbf{u}^*(t) = \operatorname{argmax}_{\mathbf{u} \in \mathrm{U}} H(t, \mathbf{x}^*, \mathbf{u}, \boldsymbol{\psi}^*) \,. \tag{1.50}$$

Для нахождения оптимального управления из условия максимума гамильтониана, необходимо в каждый момент времени знать значение вектора состояния объекта управления и вектора сопряженных переменных. Проблема в том, что для системы дифференциальных уравнений модели объекта известны начальные и конечные условия, а для сопряженных переменных существует система дифференциальных уравнений, но эта система не имеет ни начальных, ни конечных условий. Поэтому возникает краевая задача нахождения таких начальных условий для системы сопряженных переменных, что при вычислении управления на каждом шаге интегрирования из условий максимума гамильтониана вектор состояния в результате попадает в заданные терминальные условия. Данная задача формулируется как краевая задача конечномерной оптимизации, в которой необходимо найти вектор начальных условий для системы дифференциальных уравнений сопряженных переменных по критерию минимальной погрешности входа вектора состояния в конечные условия.

Появление принципа максимума стало мощным толчком к развитию теории оптимального управления и вызвало громадное количество исследований и публикаций, посвященных как изучению связи оптимального управления с классическим вариационным исчислением, так и распространению принципа максимума на более общие классы задач оптимального управления – задачи с фазовыми и смешанными ограничениями [53 - 55].

Для задач с фазовыми ограничениями необходимые условия оптимальности были получены Р.В. Гамкрелидзе [56] и опубликованы в классической монографии [22]. Позднее, А.Я. Дубовицкий и А.А. Милютин предложили и доказали новый принцип максимума для получения необходимых условий в задачах с ограничениями [57]. Несмотря на то, что он, в отличие от принципа максимума Р.В. Гамкрелидзе, был получен без априорного предположения регулярности траектории, во многих интересных случаях принцип максимума в форме Дубовицкого-Милютина вырождается. Этот эффект вырождения был открыт и изучен А.В. Арутюновым и Н.Т. Тынянским

[58] и были предложены другие варианты невырождающегося принципа максимума Дубовицкого-Милютина.

Большой вклад в развитие теории задач с фазовыми ограничениями внесли А.В. Арутюнов, С.М. Асеев, Р.Б. Винтер, Р.В. Гамкрелидзе, А.В. Дмитрук, А.Я. Дубовицкий, М.И. Зеликин, Д.Ю. Карамзин, А.Б. Куржанский, А.С. Матвеев, А.А. Милютин, Н.П. Осмоловский, Е.С. Половинкин, Г.В. Смирнов, Н.Т. Тынянский, Х. Халкин и другие ученые.

Принцип максимума считается основным инструментом определения оптимальных управлений и траекторий различных динамических объектов. Однако использование принципа максимума в классическом виде является достаточно трудоёмким. В робототехнике мы практически не встречаем новых прикладных задач, решенных с использованием принципа максимума. Основные исследования принципа максимума лежат в теоретической области, хотя изначально Л.С. Понтрягин говорил о прикладном характере принципа максимума.

Сложность самой задачи оптимального управления в робототехнике в виду нелинейности динамических объектов, необходимости учета различных фазовых ограничений, в том числе и динамических при работе с группой роботов приводят к тому, что задача оптимального управления в ее традиционной математической постановке все меньше и меньше решается при создании новых робототехнических устройств.

Дополнительная проблема заключается еще и в том, что исходя из классической постановки задачи оптимального управления мы находим управление как функцию времени. Другими словами, мы получаем разомкнутый тип управления, предполагающий выработку управляющих сигналов по времени, а не по состоянию объекта. Даже если удаётся преобразовать полученное оптимальное управление как функцию координат и параметров объекта, возникают существенные сложности с исследованием качественных свойств траекторий объекта управления. Систему управления, работающую на реальном объекте, обычно строят по принципу обратной связи.

1.3. Анализ методов решения задачи синтеза оптимального управления

Обратная связь подразумевает управление по состоянию объекта. Математически такая задача формулируется как задача синтеза управления. Задача синтеза ставится как задача поиска закона управления от фазовых переменных. В отличие от рассмотренной выше задачи оптимального управления, она носит более прикладной характер, так как здесь ищется управление как функция состояния объекта. В результате разработчик получает блок управления с обратной связью, обеспечивающий по сигналам датчиков, определяющих состояние объекта, достижение объектом цели управления при оптимальном значении критерия качества управления для любого текущего состояния объекта. В этом заключается специфика задачи синтеза управления. После решения задачи синтеза управления полученный блок управления автоматически решает задачу оптимального управления для любого текущего состояния объекта управления.

На заре создания теории управления в 60-х годах прошлого века, изучая математическую постановку задачи оптимального управления, Р. Беллман сформулировал задачу синтеза управления и представил уравнение Гамильтона-Якоби-Беллмана [59], обозначаемое в англоязычной литературе HJB (Hamilton-Jacobi-Bellman). Уравнение является дифференциальным уравнением в частных производных. Решением этого уравнения является функция Беллмана, одним из аргументов которой является вектор управления. Нахождение такого управления, которое максимизирует функцию Беллмана, является решением задачи синтеза.

$$\mathbf{u}^*(t) = \operatorname{argmax}_{\mathbf{u}\in \mathrm{U}} H(\mathbf{x}^*, \mathbf{u}, -V_{\mathbf{x}}(\mathbf{x}^*)) \,. \qquad (1.51)$$

Заметим, что уравнения в частных производных значительно сложнее обыкновенных дифференциальных уравнений и в общем случае почти никогда не имеют общего решения.

Беллман предложил численную процедуру поиска решения в виде динамического программирования [60 - 62]. Возможность генерировать

оптимальную политику управления в виде закона обратной связи по состоянию является важной особенностью метода динамического программирования.

Уравнение HJB и метод динамического программирования дают необходимые и достаточные условия для задачи оптимального управления и имеют много приложений [63, 64].

Но большинство попыток решить уравнение Беллмана в задаче синтеза системы управления рассматривают частные случаи и для них получают аналитическую формулу для функции Беллмана, например, рассматриваются линейные системы управления с квадратичным функционалом качества. Такой подход с 60-х годов активно развивался А.М. Летовым [65, 66], Р. Калманом [67], А.А. Красовским [68], В.Н. Афанасьевым [69] и другими учеными, сложившись в целое направление аналитического конструирования оптимальных регуляторов (АКОР) [70, 71]. Синтез системы управления с обратной связью по состоянию объекта по квадратичному интегральному критерию дает регулятор в виде линейных обратных связей по координатам состояния объекта управления. При этом уравнение Беллмана предлагается преобразовать в более простые для решения уравнения Риккати. При этом отметим, что используемый квадратичный функционал не учитывает возможные фазовые ограничения, накладываемые на вектор состояния объекта, что является довольно актуальным при решении прикладных задач в робототехнике. Ввиду простоты реализации такого регулятора и универсальности квадратичного критерия оптимальности, такая задача определения оптимального регулятора получила широкое распространение. Данный подход хорошо формализован, но работает только для определенного класса моделей и функционалов.

В общем случае необходимо предположить вид функции Беллмана, форма которой не всегда является очевидной. По сути, форма функции Беллмана (или функции стоимости перехода в англоязычной литературе) $V(t, \mathbf{x})$ подбирается на основе опыта, или угадывается, и далее необходимо убедиться, что это действительно решение уравнения HJB. Для задач синтеза управления с произвольными нелинейными системами и функционалами может быть

неочевидно угадывать решение уравнения в частных производных. Поэтому для более общих задач, с вычислительной точки зрения принцип максимума имеет некоторые преимущества, поскольку позволяет решать задачи, для которых уравнение HJB неразрешимо аналитически.

Довольно часто задачу синтеза системы управления заменяют задачей поиска только стабилизирующего управления. Однако заметим, что эта задача является лишь частным случаем общей задачи синтеза системы управления. Решение задачи общего синтеза системы управления обеспечивает не только устойчивость объекта в области некоторой точки равновесия в пространстве состояний, но и оптимальное по заданному критерию качества выполнение объектом цели управления.

Проблема синтеза оптимального управления является фундаментальной и доминирующей в теории управления. Вместе с тем, возникающий разрыв между результатами теоретических разработок и их практического использования увеличивается. Одной из основных причин данного несоответствия является так называемое «проклятие размерности». В результате применения динамического программирования для множества числовых значений векторов состояния мы получаем множество векторов управления, при этом не получаем никакой аналитической зависимости управления от состояния. Схожую картину мы получаем в результате применения принципа максимума для каждой точки области начальных значений (см. рис.1.2.). По сути, в общем случае, для задачи синтеза на сегодняшний день мы можем строго получить только таблицу состояний и управлений в каждый момент времени, которая должна иметь огромнейшую размерность и вряд ли применима на объекте.

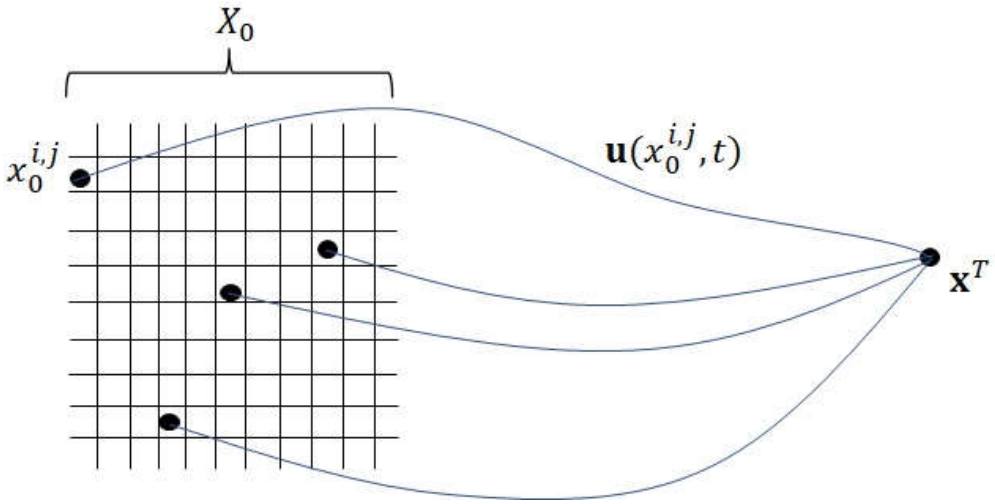

Рис.1.2. Иллюстрация к решению задачи синтеза

В идеале задачу синтеза необходимо решать как задачу *поиска закона управления по состоянию* объекта управления. Формулировка задачи общего синтеза в такой парадигме была предложена еще в [22, 72] и до настоящего времени является актуальной математической задачей.

Основные строгие аналитические подходы, рассмотренные выше, разрабатывались с тех времен, когда вычислительная техника сильно отличалась от современной и важно было уметь строить строгие аналитические решения, хотя бы для задач с моделями объектов небольшой размерности и функционалами определенных типов.

Несмотря на обширный теоретический багаж теории управления, сегодня существует широкий круг прикладных задач, не имеющих точных аналитических решений. В то же время существует объективная потребность в их решении.

С появлением современных вычислительных машин, мощность которых сегодня позволяет численно интегрировать даже очень сложные системы и решать для них различные задачи управления, открываются новые перспективы для автоматизации и численного поиска решений задач оптимального управления. В эту область и направлено настоящее диссертационное исследование.

1.4. Выводы

Стремление к построению системы управления объектом, обеспечивающей оптимальное выполнение требований к качеству ее функционирования, естественно. Современная теория оптимального управления является мощной областью науки с фундаментальным математическим основанием, кратко представленным в обзоре. Однако, задача оптимального управления позволяет получать программное управление в зависимости от времени, которое не учитывает реального состояния объекта управления. Именно задача синтеза для прикладных объектов управления имеет актуальность, так как обеспечивает управление по состоянию и учитывает неопределенности, как начальных условий, так и модели объекта управления, используемой в расчетах оптимальных режимов управления. Управление по состоянию, или управление с обратной связью, в чем и есть суть задачи синтеза, является основной задачей в управлении реальными объектами. Однако, задача синтеза в общем виде не имеет аналитических подходов к ее решению. Для сложных нелинейных систем и функционалов общего вида рассмотренные методы как правило неприменимы, поэтому актуальным и востребованным направлением является разработка современных универсальных численных подходов.

Глава 2. Принцип синтезированного оптимального управления

Рассмотренные в предыдущих подразделах диссертационного исследования задачи оптимального управления и синтеза управления являются, по сути, основными направлениями исследований в теории оптимального управления. Но в отличие от теоретических расчетов, где основной критерий качества содержится в функционале качества, практические системы управления вынуждены парировать целый ряд возникающих трудностей, связанных с неточностями используемой в расчетах модели, возникающими помехами в состоянии системы, в том числе отклонения в исходном положении, необходимость быстро реагировать на изменение реальных условий, иметь возможность пересчитывать траекторию движения на борту в режиме реального времени и т.д.

На первый взгляд может показаться, что решение задачи синтеза управления (см. раздел 1.3) как раз и направлено на преодоление указанных трудностей. Однако, на практике, особенно в области робототехники решение такой задачи не всегда актуально, поскольку, с одной стороны, решение общей задачи синтеза, включая различные фазовые ограничения, является вычислительно сложным и часто невыполним с использованием известных аналитических подходов, но при этом редко встречаются ситуации, когда условия функционирования объекта не изменяются и строго соответствую расчетным. Чаще всего мобильные робототехнические системы, а именно они являются основным объектом настоящего диссертационного исследования, функционируют в довольно изменчивых, и не всегда точно известных условиях, что требует перерасчетов оптимальных режимов управления. В таких ситуациях решение задачи синтеза становится не актуальным, и требуется разработка более гибких подходов, которые, с одной стороны, сохраняют возможность осуществлять управление по состоянию объекта, но, с другой стороны, обеспечивают возможность пересчитать оптимальные режимы управления в случае появления изменений во внешней среде.

В теории автоматического управления существует область так называемых робастных систем [73 - 78], при разработке которых возможен учет неопределенностей состояния объекта, что позволяет системе оставаться устойчивой в процессе эксплуатации. В известных подходах к робастному синтезу качество управления рассматривается с точки зрения обеспечения необходимого запаса устойчивости при наличии неопределенностей в контуре управления. Однако в этом случае приходится жертвовать оптимальностью, чтобы заложить необходимый запас устойчивости.

В идеале мы все же стремимся разрабатывать оптимальные системы, которые являются лучшими по заданному критерию качества выполнения целевой задачи. В этом случае все-таки решается задача оптимального управления.

Однако, как показано в разделе 1.2, результатом решения задачи оптимального управления является функция оптимального управления как функции управления от времени. И это ключевая проблема с точки зрения прикладной реализации полученного решения. Реализация такой функции представляет собой разомкнутый тип управления. В разомкнутой системе управления для получения желаемой реакции объекта обычно используется регулятор или исполнительное устройство, при этом обратная связь отсутствует, как показано на рис. 2.1.

Рис.2.1. Разомкнутая система управления (без обратной связи по состоянию)

В таком случае любое несовпадение во времени движения объекта и управления может привести к тому, что цель управления не будет достигнута и значение критерия качества будет отличаться от полученного при математических расчетах.

Безусловно, разомкнутое управление имеет свою прикладную ценность в определенных задачах, но в большинстве реальных приложений, в частности в области управления робототехническими системами, требуется управление по состоянию объекта, или управление с обратной связью, как показано на рис.2.2. В отличие от разомкнутой, в замкнутой системе производится измерение действительного значения выходного сигнала, по которому оценивается реальное состояние объекта управления. Измеренное значение выхода называют сигналом обратной связи, он подается на систему управления, которая затем вырабатывает сигнал управления, основываясь на состоянии объекта с целью достижения цели управления с оптимальным значением критерия качества управления.

Рис.2.2. Замкнутая система управления (с обратной связью)

Замкнутое управление по состоянию объекта управления предпочтительно по целому ряду причин.

- Неопределенность модели объекта

Модель представляет собой упрощенную версию самого объекта, охватывающую только его основные существенные свойства. С одной стороны, подробные динамические модели, точно описывающие поведение объектов управления, слишком сложны или в некоторых случаях их вообще невозможно получить. С другой стороны, наиболее распространенные алгоритмы управления были разработаны с использованием упрощенных линейных моделей. Но

производительность этих моделей может оказаться низкой, поскольку динамика робота может быть далека от линейной. Таким образом, ввиду объективно существующих отличий модели от объекта управления, или другими словами некоторых неопределенностей модели, при переносе полученных расчетных оптимальных данных на реальный объект возникают отклонения в динамике реального объекта от расчетной модели.

- Неустойчивость модели

При разомкнутом управлении на самом деле мы не можем изменить поведения самой системы. Мы лишь подаем управляющее воздействие, но при этом сама система может оставаться неустойчивой, и в таком случае не гарантируется достижения цели управления.

- Внешние возмущения

Внешние возмущения присутствуют практически всегда в реальных системах и прогнозировать их поведение очень сложно, или трудоемко. Невозможно, например, заложить в модель объекта управления такие возмущения, как порыв ветра. Но при этом такие возмущения влияют на поведение системы, вносят определенные изменения в динамику объекта, которые можно отследить и даже учесть только с помощью введения обратной связи по состоянию.

- Эффективность

Управление, учитывающее состояние объекта в каждый момент времени, может быть гораздо эффективнее, в частности энерго эффективнее, чем разомкнутое управление.

В виду всех упомянутых факторов, на практике в реальных объектах используются системы управления по состоянию, или управление с обратной связью. Такой подход позволяет преодолеть указанные трудности, включая некоторые неопределенности модели или внешние возмущения, а также можно влиять на устойчивость поведения системы, изменив расположение собственных значений матрицы системы, тем самым обеспечивая предсказуемость работы системы и достижения рассчитываемого результата.

Введение управления с обратной связью в систему дифференциальных уравнений придает системе некоторое свойство, позволяющее объекту управления достичь цели с близким к расчетному значением качества, то есть быть реализуемым. Однако, как было замечено в главе 1, задача общего синтеза для нахождения управлений по состоянию объекта для нелинейных моделей и сложных функционалов, учитывающих в том числе и фазовые ограничения, не имеет методов решения и поэтому все чаще инженеры избегают постановки таких задач, заменяя их другими задачами, более простыми в реализации.

Так при практическом проектировании систем управления существующие неопределенности, которые впоследствии вызывают расхождение между реальной траекторией объекта и полученной оптимальной, чаще всего компенсируются синтезом наиболее простой и популярной системы управления с обратной связью в виде системы стабилизации относительно оптимальной траектории [21, 79 – 83], обеспечивающей минимизацию ошибки отклонения. В таком случае, расчетная программная траектория является желаемым выходом системы. С помощью специальных датчиков в процессе работы объекта управления измеряется его фактическое положение и (или) скорость, которые передаются на вход системы управления. Фактические значения фазовых переменных сравниваются с расчетными значениями (желаемым выходом). В результате такого сравнения (вычитания) образуется сигнал, пропорциональный ошибке между желаемой и действительной реакциями системы. Целью управления является сведение ошибки к нулю. Использование этого сигнала для управления объектом приводит к появлению замкнутой системы управления, или системы управления с обратной связью, как показано на рисунке 2.3. Здесь $\mathbf{x}^*(t)$ – заданная оптимальная траектория, которая получена из решения задачи оптимального управления, $\mathbf{x}(t)$ – реальное состояние объекта, D является оператором преобразования ошибки, например, линейная передаточная функция.

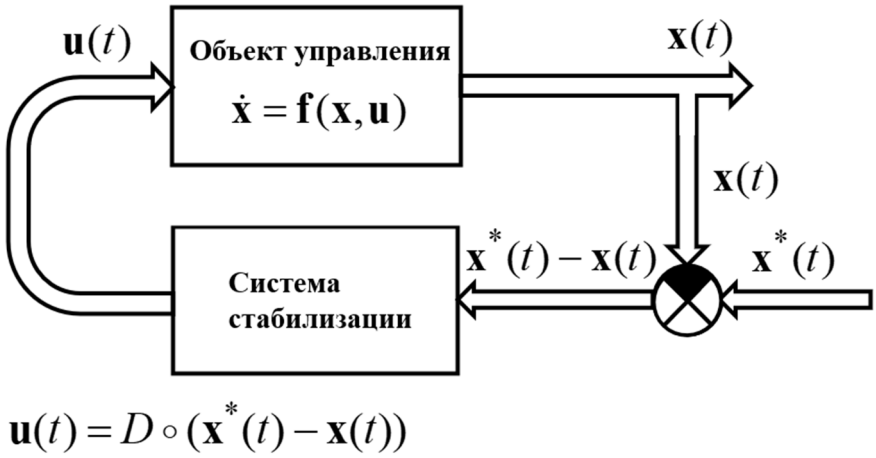

Рис. 2.3. Контур управления с системой стабилизации

Но из-за введения системы стабилизации мы можем потерять рассчитанную оптимальность по ряду причин.

- Построение системы стабилизации изменяет математическую модель объекта, и полученное управление может быть не оптимальным для новой модели.
- Ошибка движения объекта по траектории может быть как по времени, так и по положению. Например, ошибка по положению относительно траектории движения в пространстве не учитывает ошибки по времени. Обе эти ошибки могут привести к неоптимальному движению.
- Система стабилизации должна иметь ресурс управления для возврата объекта на траекторию. Это означает, что при расчете оптимального управления необходимо учитывать, что не все ресурсы управления будут доступны. А это, как правило, в расчетах не учитывается.
- Движение объекта в окрестности программной траектории может существенно отличаться от оптимального по значению функционала, как продемонстрировано в книге Л. Янга [27].

И в дополнение ко всему сказанному, задача синтеза системы стабилизации движения по программной траектории зачастую оказывается более сложной, чем исходная задача оптимального управления.

Имеется и еще одно обстоятельство, усложняющее реализацию решения задачи оптимального управления. В классической постановке задачи оптимального управления никаких дополнительных требований к математической модели объекта управления не предъявляется. Отсюда следует, что задача решается для любого объекта, в том числе неустойчивого или обладающего особыми свойствами, например, имеющего предельный цикл или неустойчивые полюса. В практической реализации неточности математической модели ведут себя по-разному в зависимости от качественных характеристик системы дифференциальных уравнений модели, поэтому разработчики стараются компенсировать их системой стабилизации с обратной связью.

На практике инженеры давно поняли описанные трудности управления неустойчивыми объектами, поэтому за счет введения системы стабилизации сначала делают объект устойчивым, а затем решают задачи управления. Известно, что объекты обладают хорошими свойствами для управления, когда их математические модели устойчивы в фазовом пространстве.

Руководствуясь анализом практически реализованных систем управления, в этой главе хотелось бы представить новый разработанный принцип синтезированного оптимального управления, позволяющий получать законы управления по принципу обратной связи, учитывающие состояния объекта. Но сначала рассмотрим вопрос реализуемости моделей объектов управления.

2.1. Реализуемость моделей объектов управления

Расчет оптимального управления сложными робототехническими системами производится на основе их математических моделей.

С точки зрения управления, робототехническая система есть некоторая нелинейная динамическая система. Нам интересно поведение этой системы в динамике, ее реакция на наши воздействия. Поэтому с математической точки зрения, объектом наших исследований является динамическая система, описываемая в общем виде системой обыкновенных дифференциальных уравнений

$$\dot{\mathbf{x}}=\mathbf{f}(\mathbf{x}, \mathbf{u}), \qquad (2.1)$$

где **x** - вектор, описывающий состояние объекта управления, **u** - управление, $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{u} \in \mathrm{U} \subseteq \mathbb{R}^m$, $m \leq n$, U - компактное множество.

В правые части дифференциальных уравнений модели (2.1) динамической системы входит свободный вектор **u**, который влияет на динамику системы. Таким образом, можно сказать, что наша математическая модель нам известна с точностью до значений вектора **u**. Решение задач управления с математической точки зрения представляет собой качественное изменение правых частей системы дифференциальных уравнений, описывающих математическую модель объекта управления. Эти изменения задаются свободным вектором управления **u** и компактом, определяющим ограничения на управление U.

Разные функции управления могут обеспечивать разные свойства системы управления или, точнее, разные свойства системы дифференциальных уравнений. Например, при решении задачи оптимального управления и нахождении функции управления как функции времени мы хотим, чтобы система дифференциальных уравнений математической модели объекта управления имела частное решение, которое из заданных начальных условий попадает в заданное терминальное состояние с оптимальным значением критерия качества.

Обратим внимание на важный момент. Модель в виде системы дифференциальных уравнений (2.1) представляет собой упрощенную версию самого объекта, охватывающую только его основные существенные свойства. Модели объективно имеют неточности. С одной стороны, динамические модели, точно описывающие поведение роботов, слишком сложны, зачастую они вообще могут отсутствовать. В то же время, работать со сложными моделями высокой размерности трудно, отсутствуют методы, пригодные для работы с такими сложными моделями. Например, наиболее распространенные методы построения систем управления были разработаны с использованием упрощенных линейных моделей. В результате при переносе полученных

расчетных оптимальных управлений на реальный объект возникают отклонения реального объекта от расчетной модели.

Нарастание ошибки между реальным объектом и расчетной моделью может приводить к необратимым последствиям. Поэтому необходимо работать с такой моделью объекта управления, ошибка которой не нарастала бы со временем функционирования, а значит модель была бы **реализуемой**.

Понятно, что не все системы реализуемы. Например, модели с разомкнутым типом управления могут не обладать свойством реализуемости. И наоборот, устойчивые системы, как правило, реализуемы. Таким образом, возникает необходимость сформулировать свойство, позволяющее определить реализуемость системы.

Введем следующее определение. Пусть $\mathbf{x}(t)$ - состояние объекта управления, полученное по его математической модели, а $\mathbf{y}(t)$ – состояние реального объекта управления, полученное в результате измерений.

Определение 2.1. Модель объекта управления является **реализуемой** на интервале $[t_0, T]$, если ее ошибка на требуемом интервале не увеличивается более, чем на некоторую заданную погрешность $\delta > 0$:

$$|\mathbf{x}(t) - \mathbf{y}(t)| \leq \delta \qquad \forall t \in [t_0, T]. \tag{2.2}$$

Предполагаем, что модель объекта, выбранная для расчетов режимов управления, описываемая системой обыкновенных дифференциальных уравнений, качественно соответствует реальному объекту. Тогда можно полагать, что реализуемость означает, что небольшие изменения в решении системы дифференциальных уравнений не приводят к потере качества.

Далее попробуем проанализировать, каким качественным свойством должна обладать система дифференциальных уравнений, описывающая объект управления, чтобы ошибка не росла на требуемом интервале реализации и каким образом можно обеспечить наличие этого свойства у модели с помощью имеющегося вектора управления.

Приведем соответствующие рассуждения.

Рассмотрим модель объекта управления в виде системы обыкновенных дифференциальных уравнений без свободного вектора управления:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}), \tag{2.3}$$

где $\mathbf{x} \in \mathbb{R}^n$.

Обыкновенное дифференциальное уравнение есть рекуррентное описание функции времени. Решением дифференциального уравнения является преобразование рекуррентной формы в функцию времени.

Компьютерное вычисление дифференциального уравнения (2.3) реализуется в следующей форме:

$$\mathbf{x}(t + \Delta t) = \mathbf{x}(t) + \Delta t \mathbf{f}\big(\mathbf{x}(t)\big), \tag{2.4}$$

где t является независимым параметром, Δt – постоянный параметр, называемый шагом интегрирования.

Правая часть уравнения (2.4) представляет собой однопараметрическое отображение пространства $\mathbb{R}^n$ в себя

$$F(\mathbf{x}, t) = \mathbf{x}(t) + \Delta t \mathbf{f}\big(\mathbf{x}(t)\big) \colon \mathbb{R}^n \to \mathbb{R}^n. \tag{2.5}$$

Согласно качественному анализу решений систем дифференциальных уравнений [84], правая часть системы дифференциальных уравнений, описываемых модель, как однопараметрическое отображение, обладает свойством сжимаемости, если

$$\rho\left(F(\mathbf{x}^a(t), t), F(\mathbf{x}^b(t), t)\right) \leq \rho\left(\mathbf{x}^a(t), \mathbf{x}^b(t)\right), \tag{2.6}$$

где $\mathbf{x}^a(t) \in \mathrm{D}$, $\mathbf{x}^b(t) \in \mathrm{D}$, $\rho(\mathbf{x}^a, \mathbf{x}^b)$ – это расстояние между двумя точками в пространстве $\mathbb{R}^n$

$$\rho(\mathbf{x}^a, \mathbf{x}^b) = \left\|\mathbf{x}^a - \mathbf{x}^b\right\|.$$

На практике нас интересует поведение объекта не на всем пространстве $\mathbb{R}^n$, а в некоторой области этого пространства. Пусть в пространстве $\mathbb{R}^n$ задан компакт D. Все интересующие нас решения системы дифференциальных уравнений (2.3) принадлежат этой области. Следовательно, для системы дифференциальных уравнений (2.3) начальные и конечные условия принадлежат этой области

$$\mathbf{x}(0) \in \mathrm{D} \subseteq \mathbb{R}^n,\ \mathbf{x}(T) \in \mathrm{D} \subseteq \mathbb{R}^n,$$

где $\mathbf{x}(T)$ – терминальная точка решения уравнения (2.3).

Если в области D для отображения (2.5) выполняется свойство (2.6), тогда можно предположить, что математическая модель (2.4) реализуема в области $\mathrm{D} \subseteq \mathbb{R}^n$.

Действительно, пусть $\mathbf{x}(t) \in \mathrm{D}$ – известное состояние системы в момент времени t, а $\mathbf{y}(t) \in \mathrm{D}$ – реальное состояние системы в тот же момент. Тогда ошибка состояния

$$\delta(t) = \rho\big(\mathbf{x}(t), \mathbf{y}(t)\big). \tag{2.7}$$

Согласно отображению (2.5):

$$\delta(t + \Delta t) = \rho\big(F(\mathbf{x}(t), t), F(\mathbf{y}(t), t)\big). \tag{2.8}$$

А согласно свойству сжимаемости (2.6):

$$\delta(t + \Delta t) \leq \delta(t). \tag{2.9}$$

Основываясь на представленных рассуждениях, введем следующее определение.

Определение 2.2. Система дифференциальных уравнений является реализуемой, если эта система как однопараметрическое отображение обладает свойством сжимаемости в области реализации.

Реализуемость на интервале $[t_0, T]$ модели объекта, имеющего в правых частях системы дифференциальных уравнений свободный вектор управления (2.1), может быть достигнута различными способами.

Прежде всего, реализуемость модели объекта управления на некотором интервале $[t_0, T]$ может достигаться путем получения моделей высокой точности. Это отдельная сложная, но актуальная задача получения высокоточной модели, которая на достаточном, но ограниченном интервале времени не будет превышать величину ошибки с реальным объектом на допустимую величину, и в то же время модель не будет перегружена и будет учитывать только те элементы динамики, которые влияют на достижение цели управления.

Разработка моделей, описывающие интересующую систему, является довольно трудоемким процессом, требующим подробных специальных знаний об объекте управления. Для слабо изученных систем ручное выведение модели часто вообще невозможно. Современным перспективным направлением здесь является получение модели путем ее идентификации.

В задаче идентификации математическая модель объекта управления не известна полностью или частично, но исследователь имеет реальный объект управления или его симулятор. Необходимо найти неизвестную многомерную функцию, описывающую динамику объекта. Пространство входных векторов этой функции есть пространство допустимых управлений для этого объекта. Идентификация системы позволяет получать довольно точные, но при этом компактные модели, учитывающие определенные нужные свойства системы. В подавляющем большинстве методов реализуется так называемая параметрическая идентификация [85, 86]. Сначала выбирается определенная структура модели, которая считается подходящей для описания данного объекта. Далее проводится идентификационный эксперимент, в котором измеряются входные и выходные сигналы, а затем метод идентификации реализует настройку параметров модели в соответствии с некоторыми адаптивными законами, чтобы реакция модели на входной сигнал могла приблизительно соответствовать отклику реальной системы к тому же входному воздействию. Чаще всего используют идентификацию объектов с помощью линейных систем [87, 88], поскольку для них легко определить влияние различных входных сигналов на выход. Хотя линейные модели привлекательны по многим причинам, они имеют свои ограничения, тем более что все физические системы в той или иной степени нелинейны, и во многих случаях линейные модели не подходят для представления этих систем.

В связи с этим в настоящее время возрос значительный интерес к методам идентификации нелинейных систем [89 – 91], особенно с помощью методов машинного обучения на основе нейронных сетей [92 – 94]. Примеры таких моделей, идентифицированных искусственными нейронными сетями, получены

и исследовались автором в работах [95, 96]. При этом заметим, что с помощью нейронных сетей также реализуется параметрический подход к идентификации. В последние годы активно исследуется вопрос структурно-параметрической идентификации на основе символьной регрессии [97 – 99], которые позволяют искать не только параметры системы, но и саму структуру. Данная область исследования возникла не так давно и результаты пока не превзошли искусственные нейронные сети, но перспектива безусловно хорошая, поскольку получаемые модели имеют интерпретируемую человеком форму, в отличие от нейронных сетей.

В таком случае, используя модели высокой точности, управление объектом может быть реализовано на некотором конечно временном интервале по разомкнутому типу, поскольку предполагается, что программное управление, рассчитанное с помощью высокоточной модели, на требуемом интервале времени будет отработано объектом с достаточной точностью. Однако, как уже отмечалось, получение таких достаточно точных моделей трудоемко и не всегда возможно. И кроме того, даже самая высокоточная модель не учитывает неопределенности начального состояния или условий функционирования.

Для модели объекта управления (2.1), описываемой системой дифференциальных уравнений, имеющих свободный вектор управления в правых частях, можно обеспечить свойство реализуемости на некотором интервале $[t_0, T]$ за счет введения специального управления $\mathbf{u} = \mathbf{g}(\mathbf{x}, t)$, придающего модели свойство реализуемости.

С инженерной точки зрения введение системы стабилизации как функции управления в обратной связи как раз и обеспечивает модели объекта свойство реализуемости, поскольку позволяет регулировать ошибку так, чтобы она не росла более, чем на некоторую заданную величину $\delta > 0$.

С математической точки зрения, вводя стабилизирующее управление в модель объекта (2.1) по состоянию $\mathbf{g}(\mathbf{x}, t)$ в обратной связи, мы изменяем дифференциальные уравнения самой системы объекта $\dot{\mathbf{x}} = \mathbf{f}\big(\mathbf{x}, \mathbf{g}(\mathbf{x}, t)\big)$ так, что вокруг некоторого частного решения системы $\mathbf{x}(t, \mathbf{x}^0)$ появляется некоторая

область, такая, что другие траектории, попадающие в эту область, не покидали бы ее:

если

$$\|\mathbf{x}(t', \mathbf{x}^0) - \mathbf{x}(t', \mathbf{x}^{0*})\| \leq \sigma, \tag{2.10}$$

где $t' > 0$, $\sigma > 0$,

тогда $\exists\ \alpha > 0$ такое, что для любого $\varepsilon^+ > 0$

$$\|\mathbf{x}(t' + \alpha, \mathbf{x}^0) - \mathbf{x}(t' + \alpha, \mathbf{x}^{0*})\| \leq \varepsilon^+. \tag{2.11}$$

Для этих целей на практике в подавляющем большинстве случаев в модели реальных систем в обратную связь вводятся дополнительные системы стабилизации и контуры управления. Введение стабилизирующего управления с обратной связью в систему дифференциальных уравнений придает системе некоторое свойство, позволяющее достичь цели с оптимальным значением качества и быть реализуемым.

Предположим, что система (2.3) в окрестности области D имеет одну устойчивую точку равновесия, а других точек равновесия в этой окрестности нет

$$\mathbf{f}(\tilde{\mathbf{x}}) = 0, \tag{2.12}$$

$$\det\left(\lambda\mathbf{E} - \mathbf{A}(\tilde{\mathbf{x}})\right) = \lambda^n + a_{n-1}\lambda^{n-1} + \cdots + a_1\lambda + a_0 = \prod_{j=1}^{n}\left(\lambda - \lambda_j\right) = 0, \tag{2.13}$$

где $\mathbf{E}$ – единичная матрица размерностью $n \times n$,

$$\lambda_j = \alpha_j + i\,\beta_j, \tag{2.14}$$

$$\mathbf{A} = \frac{\partial \mathbf{f}(\tilde{\mathbf{x}})}{\partial \mathbf{x}}, \tag{2.15}$$

$$\alpha_j < 0,\ i = \sqrt{\{-1\}}, j = 1, \ldots, n.$$

Покажем, что, если для системы (2.3) существует область *D*, включающая точку устойчивого равновесия (2.12) – (2.15), то система (2.3) реализуема.

Действительно, согласно теореме Ляпунова об устойчивости по первому приближению, тривиальное решение дифференциального уравнения (2.3) устойчиво

$$\mathbf{x}(t) = \tilde{\mathbf{x}} = \text{const}.$$

Это означает, что если любое решение начинается из другой начальной точки $\mathbf{x}^a$, то оно будет асимптотически приближаться к устойчивому решению

$$\rho(\mathbf{x}(t+\Delta t, \mathbf{x}^a), \tilde{\mathbf{x}}) \leq \rho(\mathbf{x}(t, \mathbf{x}^a), \tilde{\mathbf{x}}),$$

где $\mathbf{x}(t, \mathbf{x}^a)$ - решение дифференциального уравнения (2.3) из начальной точки $\mathbf{x}^a$.

То же верно и для другого начального условия $\mathbf{x}^b$

$$\rho(\mathbf{x}(t+\Delta t, \mathbf{x}^b), \tilde{\mathbf{x}}) \leq \rho(\mathbf{x}(t, \mathbf{x}^b), \tilde{\mathbf{x}}).$$

Отсюда следует, что область D имеет точку $\tilde{\mathbf{x}}$ сжимающегося отображения, поэтому расстояние между решениями $\mathbf{x}(t, \mathbf{x}^a)$ и $\mathbf{x}(t, \mathbf{x}^b)$ также стремится к нулю или

$$\rho\left(\mathbf{x}(t+\Delta t, \mathbf{x}^a), \mathbf{x}(t+\Delta t, \mathbf{x}^b)\right) \leq \rho\left(\mathbf{x}(t, \mathbf{x}^a), \mathbf{x}(t, \mathbf{x}^b)\right).$$

Таким образом, для получения реализуемого решения задачи оптимального управления необходимо построить такую функцию управления $\mathbf{g}(\mathbf{x}, t)$, которая заставляет объект достигать заданной цели с оптимальным значением критерия качества и удовлетворяет свойству реализуемости.

На практике инженеры давно поняли трудности управления неустойчивыми объектами, поэтому сначала делают объект устойчивым относительно определенной точки пространства состояний, а затем располагают устойчивые точки на полученной в результате решения задачи оптимального управления программной траектории. В результате объект перемещается по траектории в целевую позицию, следуя по этим точкам от одной точки к другой [21, 100 - 805]. Такой подход имеет наибольшую популярность в современной робототехнике, поскольку просто в реализации и легко адаптивен к изменению отрабатываемой траектории.

Основным недостатком такого подхода при движении объекта управления по устойчивым точкам траектории является то, что даже если эта траектория получена в результате решения задачи оптимального управления, то само движение не обязательно будет оптимальным, например, в случае, если в функционал качества входит критерий быстродействия. Действительно, для

обеспечения оптимальности необходимо двигаться по траектории с определенной скоростью, но при приближении к точке устойчивого равновесия скорость объекта управления стремится к нулю.

Необходимо, чтобы движение по устойчивым точкам тоже исходило из решения задачи оптимального управления и соотносилось с функционалом качества.

Для получения таких управлений в настоящем диссертационном исследовании предложен новый ***принцип синтезированного оптимального управления***.

Принцип синтезированного оптимального управления представляет собой математический подход к решению задачи оптимального управления в терминах функционала качества. Подход достаточно универсален, так как не привязан к конкретным свойствам модели объекта управления или типу траектории, не требует ручного выбора каналов управления, подбора и настройки регуляторов. Разработка системы управления происходит в автоматическом режиме с использованием современных вычислительных методов машинного обучения.

2.2. Принцип синтезированного оптимального управления

Согласно *принципу синтезированного оптимального управления*, решение задачи оптимального управления производится для объекта стабилизированного относительно точки равновесия в пространстве состояний.

Основная идея подхода состоит в том, что ищется функция управления, при которой система дифференциальных уравнений, описывающих объект, всегда будет иметь устойчивую точку равновесия в пространстве состояний. При этом функция управления содержит параметры, влияющие на положение точки равновесия. Следовательно, управление объектом осуществляется за счет изменения положения точки равновесия, а именно оптимального ее расположения на заданных интервалах.

Для принципа синтезированного оптимального управления можно провести параллель с общими законами устройства природы. Можно полагать,

что этот принцип соответствует парадигме баланса, высказанной русским нейрофизиологом, лауреатом Нобелевской премии Иваном Петровичем Павловым, для балансировки вызовов новых сложных задач управления.

В своих «Лекциях о работе полушарий головного мозга» И.П. Павлов сказал [106]: «Каждая материальная система до тех пор может существовать как данная отдельность, пока ее внутренние силы притяжения, сцепления и т. д. уравновешиваются с внешними влияниями, среди которых она находится. Это относится ко всякому простому камню, как и к сложнейшему химическому веществу. Точно то же надо представлять себе и относительно организма. Как определенная замкнутая вещественная система он может существовать только до тех пор, пока он каждый момент уравновешивается с окружающими условиями. Как только это уравновешивание серьезно нарушается, он перестает существовать как данная система». Павлов предполагал, что все сложные физиологические организмы стремятся к уравновешиванию внешних воздействий. В этом отношении можно полагать, что новые современные вычислительные методы появились в противовес новым сложным вычислительным задачам.

Согласно принципу синтезированного оптимального управления необходимо найти такую функцию управления, чтобы система дифференциальных уравнений, описывающих модель объекта управления, всегда имела устойчивую точку равновесия в пространстве состояний. Вместе с тем в функцию управления вводится вектор параметров $\mathbf{q}^*$. Значение этого вектора параметров влияет на положение точки равновесия в пространстве состояний

$$\mathbf{u} = \mathbf{g}(\mathbf{x}, \mathbf{q}^*), \tag{2.16}$$

где $\mathbf{q}^*$ – вектор параметров.

Функция управления (2.27) обеспечивает наличие у системы

$$\dot{\mathbf{x}} = \mathbf{f}\big(\mathbf{x}, \mathbf{g}(\mathbf{x}, \mathbf{q}^*)\big) \tag{2.17}$$

точки равновесия

$$\mathbf{f}\left(\mathbf{x}^*(\mathbf{q}^*), \mathbf{g}(\mathbf{x}^*(\mathbf{q}^*), \mathbf{q}^*)\right) = 0, \quad (2.18)$$

где $\mathbf{x}^*(\mathbf{q}^*)$ – вектор координат точки равновесия, зависящий от вектора параметров $\mathbf{q}^*$. Система (2.18) удовлетворяет условиям (2.13) - (2.15) в точке $\mathbf{x}^*(\mathbf{q}^*)$.

Алгоритмически, согласно принципу, реализуется последовательно решение двух задач:

- задачи синтеза системы стабилизации для обеспечения устойчивости объекта относительно точки в пространстве состояний
- и задачи оптимального управления, заключающееся в параметрической оптимизации точек пространства состояний, относительно которых синтезированная на первом этапе система управления должна обеспечивать устойчивость. Последовательное переключение найденных точек в пространстве состояний обеспечивает движение объекта от начального состояния до конечного с учетом фазовых ограничений и оптимального значения заданного критерия качества.

Приведем их математические постановки.

2.2.1. Этап1. Задача численного синтеза системы стабилизации.

Представим постановку задачи численного синтеза системы стабилизации.

Задана математическая модель объекта управления:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}). \quad (2.19)$$

Задано множество начальных условий:

$$\mathrm{X}_0 = \{\mathbf{x}^{0,1}, \dots, \mathbf{x}^{0,K}\}, \quad (2.20)$$

где $\mathbf{x}^{0,j} \in \mathrm{X}_0^n \subseteq \mathbb{R}^n$, $\left\|\mathbf{x}^{0,i} - \mathbf{x}^{0,j}\right\| \leq \delta$, $i, j \in \{1, \dots, K\}$, δ - заданная величина, определяющая размер области X_0 начальных условий.

Задано терминальное состояние в виде целевой точки стабилизации:

$$\mathbf{x}^* = [x_1^* \dots x_n^*]^T. \quad (2.21)$$

Задан критерий качества

$$J_1 = \sum_{i=1}^{K} (p_1 \left\|\mathbf{x}(T_i, \mathbf{x}^{0,i}) - \mathbf{x}^*\right\| + T_i) \to \min_{\mathbf{u} \in \mathrm{U}}, \quad (2.22)$$

где p_1 - весовой коэффициент, T_i - время достижения терминального состояния (2.21) из начального положения $\mathbf{x}^{0,i}$

$$T_i = \begin{cases} t, \text{если } t < t^+ \text{ и } \left\|\mathbf{x}(t, \mathbf{x}^{0,i}) - \mathbf{x}^*\right\| \le \varepsilon, \\ t^+ - \text{ иначе,} \end{cases} \tag{2.23}$$

$i = 1, \ldots, K$, ε и t^+ - заданные положительные величины.

Для решения задачи синтеза системы стабилизации необходимо найти функцию управления в форме

$$\mathbf{u} = \mathbf{g}(\mathbf{x}, \mathbf{x}^*) \in \mathrm{U}, \tag{2.24}$$

доставляющую минимум функционалу J_1.

Такое представление задачи стабилизации в общей математической формулировке позволит использовать универсальные методы машинного обучения для автоматического поиска многомерная функция управления, доверив решение задачи машине.

В результате решения задачи синтеза системы стабилизации (2.19) – (2.24) в пространстве состояний объекта появляется устойчивая точка равновесия, попадая в окрестность которой, объект стремится к ней с качеством, определяемым заданным функционалом (2.22), т.е. мы не просто делаем объект устойчивым относительно некоторой точки в пространстве состояний, а делаем это с заданным качеством. Обратим внимание, что функционал (2.22) включает в себя две составляющие, а именно, быстродействие и точность. Все дополнительные условия, учитывающие среду функционирования объекта в виде фазовых ограничений, будут учитываться на втором этапе решения задачи оптимального управления для стабилизированного объекта.

С введенной системой стабилизации вместо свободного вектора управления объект в каждый момент времени имеет устойчивую точку равновесия, но в процессе управления в данной точке объект не находится, он движется по направлению к этой точке, и в процессе движения положение этой точки меняется на новое, пока не будет достигнуто целевое терминальное состояние. Решение задачи оптимального управления состоит в том, чтобы найти оптимальное положение точек равновесия, переключение которых через

заданный интервал времени обеспечит движение объекта из заданного начального состояния в терминальное с оптимальным значением критерия качества.

2.2.2. Этап 2. Задача оптимизации положения точек равновесия.

На втором этапе решается следующая задача оптимального управления:

$$\begin{gathered}\dot{\mathbf{x}} = \mathbf{f}\big(\mathbf{x}, \mathbf{g}(\mathbf{x}, \mathbf{x}^*)\big), \\ \mathbf{x}^* \in \mathrm{X}_1 \subseteq \mathbb{R}^n, \\ \mathbf{x}(0) = \mathbf{x}^0, \\ \mathbf{x}(T) = \mathbf{x}^f, \\ J = \int_0^T f_0 \ \big(\mathbf{x}, \mathbf{g}(\mathbf{x}, \mathbf{x}^*)\big) dt \to \min_{\mathbf{x}^* \in \mathrm{X}_0}.\end{gathered} \tag{2.25}$$

Необходимо найти функцию управления как функцию времени

$$\mathbf{x}^* = \mathbf{v}^*(t), \tag{2.26}$$

такую, что система

$$\dot{\mathbf{x}} = \mathbf{f}\big(\mathrm{x}, \mathbf{g}(\mathbf{x}, \mathbf{v}^*(t),)\big) \tag{2.27}$$

будет иметь частное решение $\mathbf{x}(t, \mathbf{x}^0)$, которое из данного начального условия $\mathbf{x}^0$ достигнет заданного конечного условия $\mathbf{x}(T, \mathbf{x}^0) = \mathbf{x}^f$, $T < t^+$, с оптимальным значением заданного критерия качества

$$J(\mathbf{v}^*(t)) = \int_0^T f_0 \left(\mathbf{x}(t, \mathbf{x}^0) , \mathbf{g}\big(\mathbf{v}^*(t), \mathbf{x}(t, \mathbf{x}^0)\big)\right) dt \to \min_{\mathbf{x}^* \in \mathrm{X}_1}. \tag{2.28}$$

В общем случае функция управления (2.26) может быть некоторая функция времени $\mathbf{x}^* = \mathbf{v}^*(t)$. Свойства этой функции и методы ее нахождения исследовались автором в работе [107], и это направление может быть в дальнейшем исследовано дополнительно.

В прикладных задачах диссертационного исследования, функция управления (2.26) рассматривается в форме кусочно-постоянной функции времени путем разбиения временного отрезка $[0; t^+]$ на интервалы Δt. Таким образом, в задаче оптимального управления необходимо найти значения координат точки устойчивости для каждого интервала через определение

оптимального значения вектора параметров $\mathbf{q}^*$, доставляющего минимум функционалу качества (2.28)

$$\mathbf{v}^*(t) = \mathbf{q}^{*,j}, \text{если } t \in [(j-1)\Delta t;\ j\Delta t), \quad j = 1, \dots, k, \tag{2.29}$$

где k - количество интервалов,

$$k = \left\lfloor \frac{t^+}{\Delta t} \right\rfloor + 1. \tag{2.30}$$

В таком случае, на втором этапе решаем задачу оптимального управления как задачу конечномерной оптимизации вектора параметров $\mathbf{q}^* = [\mathbf{q}^{*,1}, \dots, \mathbf{q}^{*,k}]^T$.

2.3. Адаптивное синтезированное оптимальное управление

Дополнительно в зависимости от степени неопределенности технической задачи, на решение которой направлен данный подход, на втором этапе можно рассмотреть случай, когда есть некоторая неопределенность в начальных условиях. Поэтому задачу оптимального управления, как задачу оптимизации положения точек равновесия, можно рассматривать не из одного начального состояния, как в постановке (2.25), а из области начальных значений. Такой подход назван принципом адаптивного синтезированного оптимального управления [108].

Пусть задано следующее множество начальных значений

$$\widetilde{\mathrm{X}}_0 = \{\mathbf{x}^{0,1}, \dots, \mathbf{x}^{0,j}, \dots, \mathbf{x}^{0,M}\}, \tag{2.31}$$

где M - заданное количество точек начальных значений.

С учетом найденной на первом этапе функции стабилизации (2.24), множества начальных значений (2.31) и с включением точности попадания в терминальное состояние в критерий качества, функционал для задачи оптимального управления положением точек равновесия на основе адаптивного принципа синтезированного оптимального управления имеет следующий вид:

$$J_2 = \sum_{i=1}^{M} \left(\left\| \mathbf{x}\left(T_i, \mathbf{x}^{0,i}\right) - \mathbf{x}^f \right\| + \int_0^{T_i} f_0(\mathbf{x}, \mathbf{g}(\mathbf{x}, \mathbf{q}^*))dt \right) \to \min_{\mathbf{q}^* \in \mathrm{Q}}, \tag{2.32}$$

где Q - компакт в пространстве параметров, T_i определяется из соотношения (2.23) с заменой $\mathbf{x}^*$ на $\mathbf{x}^f$.

2.4. Расширенная постановка задачи оптимального управления

Управление, полученное на базе принципа синтезированного оптимального управления, обеспечивает существование свойства реализуемости, которое приобретает модель за счет введения, в частности, стабилизирующего управления в обратную связь.

Желательно, чтобы требование реализуемости вытекало бы непосредственно из постановки задачи оптимального управления. Такая постановка задачи оптимального управления в ее расширенном варианте была представлена А.И. Дивеевым [109, 110], где вводится дополнительное требование к оптимальной траектории. Оптимальная траектория должна иметь непустую окрестность, обладающую свойством притяжения. Выполнение этих требований обеспечивает получение не только оптимального с точки зрения функционала управления, но и реализуемого на реальных объектах. Принцип синтезированного оптимального управления является одним из подходов к решению расширенной постановки задачи оптимального управления, предложенной Дивеевым А.И. В работах [110, 111] автором представлены возможности применения принципа синтезированного оптимального управления для решения задачи оптимального управления в ее расширенной постановке.

Рассмотрим расширенную постановку задачи оптимального управления, решение которой реализуемо на объекте. Для этого, учитывая возможные неопределенности модели объекта или возмущения начальных условий, необходимо сформулировать задачу оптимального управления так, чтобы все возможные траектории находились в некоторой заданной области от оптимальной траектории и достигали конечного состояния.

Задана математическая модель объекта управления

$$\dot{\mathbf{x}}=\mathbf{f}(\mathbf{x}, \mathbf{u}), \tag{2.33}$$

где $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{u} \in \mathrm{U} \subseteq \mathbb{R}^m$, $\mathbf{f} = [f_1(\mathbf{x}, \mathbf{u}) \ldots f_n(\mathbf{x}, \mathbf{u})]^T$.

Заданы начальные условия

$$\mathbf{x}(0) = \mathbf{x}^0 \in \mathbb{R}^n. \tag{2.34}$$

Задано целевое терминальное состояние

$$\mathbf{x}(T) = \mathbf{x}^f \in \mathbb{R}^n, \tag{2.35}$$

где T – время достижения терминальных условий, это время может быть не задано, но ограничено, $T \le t^+$, t^+ – заданное предельное время процесса управления.

Задан критерий качества:

$$J_1 = \int_0^T f_0(\mathbf{x}, \mathbf{u})dt \to \min_{\mathbf{u}\in \mathrm{U}}. \tag{2.36}$$

Тогда для решения задачи оптимального управления (2.33) – (2.36) необходимо найти функцию управления в виде

$$\mathbf{u} = \mathbf{g}(\mathbf{x}, t) \in \mathrm{U}, \tag{2.37}$$

такую, что, если эту функцию управления $\mathbf{g}(\mathbf{x}, t)$ подставить в правую часть математической модели объекта управления, то полученная система уравнений

$$\dot{\mathbf{x}} = \mathbf{f}\big(\mathbf{x}, \mathbf{g}(\mathbf{x}, t)\big), \tag{2.38}$$

должна обладать следующими свойствами:

А) частное решение $\mathbf{x}(t, \mathbf{x}^0)$ системы (2.38) из заданного начального состояния $\mathbf{x}^0$ достигает терминального состояния $\mathbf{x}(T, \mathbf{x}^0) = \mathrm{x}^f$, $T \le t^+$ с оптимальным значением критерия качества;

В) оптимальное решение $\mathbf{x}(t, \mathbf{x}^0)$ системы (2.38) будет иметь окрестность $\Delta(t) > 0$ такую, что если для любого другого частного решения $\mathbf{x}(t, \mathbf{y})$ системы (2.38) из другого начального состояния $\mathbf{y} \in X_0 \subseteq \mathbb{R}^n$ в момент времени t', $0 \le t' \le t^+$

$$\|\mathbf{x}(t', \mathbf{y}) - \mathbf{x}(t', \mathbf{x}^0)\| \le \Delta(t'), \tag{2.39}$$

то $\forall\, t, t' \le t \le t_f$, это частное решение не покидает окрестности оптимального

$$\|\mathbf{x}(t, \mathbf{y}) - \mathbf{x}(t, \mathbf{x}^0)\| \le \Delta(t),\ t' \le t \le t_f; \tag{2.40}$$

С) окрестность $\Delta(t)$ сужается вблизи терминального состояния. Это означает, что существует η-окрестность терминального состояния $\eta \le \Delta$ такая,

что для любого частного решения $\mathrm{x}(t, \mathrm{x}^*)$ в Δ-окрестности выполняются следующие условия:

если $t'' < t^+$

$$\|x(t'', x^*) - x(t', x^0)\| \leq \eta, \tag{2.41}$$

тогда $\forall t\ , t'' \leq t \leq t^+$

$$\|\mathrm{x}(t,\ \mathrm{x}^*) - \mathrm{x}(t,\ \mathrm{x}^0)\| \leq \|\mathrm{x}(t'',\ \mathrm{x}^*) - \mathrm{x}(t'',\ \mathrm{x}^0)\|, \tag{2.42}$$

и $\exists\ t \leq\ t^+$, когда

$$\|\mathrm{x}(t,\ \mathrm{x}^*) - \mathrm{x}(t,\ \mathrm{x}^0)\| \leq \varepsilon_1, \tag{2.43}$$

где ε_1 - заданная точность достижения терминального состояния.

Существование окрестности оптимального решения во многих случаях может ухудшить значение функционала. Например, в задаче с фазовыми ограничениями, которыми являются препятствия на пути движения объекта управления к конечному состоянию, оптимальная траектория часто проходит по границе препятствия. Такая траектория не будет иметь окрестности, поэтому в этом случае необходимо найти другую траекторию, которая не будет оптимальной согласно классической постановке задачи оптимального управления, но допускает вариации начальных состояний при небольшом допустимом изменении величины функционала.

Дополнительные требования (2.39) – (2.43) означают, что оптимальная траектория, полученная в результате решения расширенной задачи оптимального управления (2.33) – (2.43), будет обладать дополнительным специальным свойством. Оно будет притягивать ближайшие частные решения: если решение попадет в некоторую дельта-трубку вблизи оптимального решения, то оно не покинет эту трубку и, более того, в окрестности целевой конечной точки сжимается к ней. Таким образом, полученное оптимальное управление будет нечувствительным к некоторым неточностям модели или возмущениям, что позволит реализовать его непосредственно на борту.

Заметим, что этому требованию соответствуют оптимальные решения, устойчивые по Ляпунову [112, 113]. Можно даже сказать, что в расширенной постановке мы ищем оптимальное управление, обеспечивающее устойчивость движения по Ляпунову. Однако сложность заключается в том, что не существует общих рекомендаций или общих подходов, как найти такое решение. Существующие работы в основном посвящены анализу, а не синтезу таких движений [114 - 117]. Чаще всего обеспечивается устойчивость в точке, а не устойчивость движения, поскольку сложность здесь в том, что устойчивость движения должна быть обеспечена во времени. Обычно инженеры делают устойчивость в точке, расположенной на оптимальной траектории. Однако устойчивость должна быть обеспечена и по критерию качества, ведь, например, вы можете двигаться в точку по прямой линии, а можете двигаться по спирали. Оба движения устойчивы, поэтому необходимо учитывать и качество движения.

Соблюдение как устойчивости, так и качества движения обеспечивается введенными требованиями (2.39) – (2.43).

Заметим, что решение общей задачи синтеза, когда возможно решить ее для определенного класса моделей и функционалов, является одним из способов решения представленной расширенной постановки задачи, получая оптимальное управление, удовлетворяющее введенным дополнительным требованиям. Но, по-прежнему, необходимо разработать какой-то общий подход, который можно применять для объектов и функционалов общего вида. Кроме того, как уже упоминалось, решение задачи синтеза – это вычислительно затратный процесс, и при изменяемости условий функционирования не всегда оправдан.

Согласно принципу синтезированного оптимального управления, в первую очередь, обеспечивается стабилизация объекта, т.е. в фазовом пространстве возникает точка равновесия. В окрестности точки равновесия фазовые траектории сжимаются, и это свойство определяет реализуемость системы. В этом главная особенность данного подхода по сравнению с задачей оптимального управления, описанной в разделе 1.2, где в результате разработчик получает разомкнутую систему управления.

Для обеспечения этого свойства необходимо численно решить задачу синтеза системы стабилизации, чтобы получить выражения для управления и подставить их в правые части модели объекта. Здесь может возникнуть закономерный вопрос: если все-таки необходимо решать задачу синтеза, может быть, лучше рассматривать задачу непосредственно как задачу общего синтеза (как это описано в разделе 1.3)? Как отмечалось ранее, задача общего синтеза очень сложна, в том числе и с вычислительной точки зрения. Она должна быть решена заранее с учетом всех возможных фазовых ограничений. Однако в большинстве прикладных систем это не всегда возможно. Как правило, объект управления работает в динамической среде и крайне важно, чтобы оптимальный путь можно было пересчитать быстро или вообще рассчитать на борту. Именно это и позволяет сделать подход, основанный на принципе синтезированного оптимального управления. Задача синтеза системы стабилизации решается заранее, а уже оптимальное положение точек равновесия, как параметров системы управления, может быть рассчитано либо заранее, либо в реальном времени на борту.

В итоге, благодаря такому синтезированному подходу к решению задачи оптимального управления, мы, с одной стороны, обеспечиваем свойство реализуемости найденному управлению, и, с другой стороны, переводим задачу из класса задач бесконечномерной оптимизации к конечномерной, обеспечивая тем самым возможность применения для ее решения широкого набора численных методов оптимизации.

2.5. Методы решения задачи оптимального управления на основе принципа синтезированного оптимального управления

В соответствии с представленной формальной постановкой метода синтезированного оптимального управления для его реализации потребуется применение сначала методов синтеза систем управления, а затем методов оптимизации.

2.5.1. Методы синтеза системы стабилизации

Поскольку задача синтеза системы стабилизации чрезвычайно важна в теории управления, уже накоплен большой багаж методов ее решения.

Наиболее эффективные методы стабилизации по состоянию разработаны для линейных систем [118 - 125], среди которых наиболее популярны методы синтеза, основанные на использовании частотных методов [126 - 128], модальном управлении [129, 130] и на основе использования матричного уравнения типа Риккати [131 - 133]. Однако, как правило, полученные алгоритмы либо преследуют решение локальной задачи, например, регулирование выхода, либо математические модели объектов должны иметь некоторую определенную структуру.

Среди более общих аналитических подходов к синтезу управления нелинейными системами отметим популярный сегодня метод бэкстэппинга, разработанный Кокотовичем [134, 136]. Суть этого метода заключается во включении в функцию управления некоторых нелинейностей на основе анализа правых частей дифференциальных уравнений с целью их компенсации и получения знакопостоянной функции Ляпунова для замкнутой системы управления, например, с четными степенями компонент вектора состояния и с одним и тем же знаком. Однако этот метод тоже зависит от модели объекта управления и особенно хорошо работает для каскадных систем, в которых одни координаты вектора состояния являются управляющими для других координат, например, в некоторых летательных аппаратах перемещение в пространстве управляется угловым положением. Применение этого метода достаточно эффективно для систем низкого порядка.

Близким по своей сути является метод аналитического проектирования агрегированных регуляторов (АДАР), разработанный А.А. Колесниковым [137, 138]. Метод заключается во введении агрегированных переменных, описывающих цель управления, например конечное состояние. Эти переменные вводятся в функционал и затем при составлении уравнения Беллмана по ним берется производная по времени. При аналитическом расчете производных

агрегированные переменные включают в себя правые части модели объекта управления, поэтому зависят от вектора управления. Тогда мы получим систему нелинейных уравнений, число которых почти всегда равно размерности вектора состояния. Эти уравнения включают вектор управления. Решая эти уравнения относительно вектора управления, мы получаем функцию управления как функцию координат пространства состояний. Есть задачи, в которых метод оказывается эффективным; однако заметим, что, во-первых, вектор управления, как правило, имеет размерность меньшую, чем вектор состояния, поэтому система нелинейных уравнений имеет множество решений относительно управления, а во-вторых, подобно бэкстэппингу, является ручным методом, который не поддается машинной автоматизации.

Существуют и другие аналитические методы решения задачи синтеза, например, методы основанные на применении функции Ляпунова [139, 140], но все известные методы аналитического синтеза подходят для определенных типов модели объекта управления.

Сегодня в большинстве приложений, как правило, специалисты решают задачу синтеза управления с помощью так называемого технического синтеза. По модели они определяют каналы управления, т.е. определяют, какие компоненты вектора управления влияют на компоненты вектора состояния. Далее в эти каналы вставляются регуляторы, чаще всего ПИД- или ПИ-регулятор, или какой-то другой регулятор, возможно даже нелинейный. Затем с помощью компьютера находятся параметры этих регуляторов. Большинство систем управления были построены с использованием такого технического подхода. В настоящий период этот метод применяется и для роботов, но это, по сути, ручной труд разработчика.

Таким образом, на сегодняшний день в подавляющем большинстве случаев задача синтеза управления решается аналитически или технически с учетом специфики математической модели. Сегодня для решения задачи синтеза для нелинейных динамических объектов различной сложности могут быть применены современные численные методы машинного обучения [141 - 146].

Машинное обучение базируется на идее, что вычислительные системы способны демонстрировать поведение, которое не было в них явно запрограммировано, они могут выявлять закономерности или функциональные зависимости и самостоятельно вырабатывать решения. Во многих научных дисциплинах, если не во всех, основная задача исследований состоит в нахождении функциональной зависимости между определенными значениями параметров, характеризующих свойства исследуемого объекта. Важность задачи поиска математического выражения некоторой функции подтверждается и мировыми тенденциями. В опубликованной в 2023 году статье Nature [147] команда Google DeepMind представила первые результаты, достигнутые с помощью применения одного из методов символьной регрессии, а именно линейного генетического программирования и большой языковой модели (LLM) для поиска функций, записанных в компьютерном коде. FunSearch построила более эффективный алгоритм, чем удавалось математикам, для поиска специальной последовательности, известной как оценка «шапочного множества». Однако, пока рано говорить об универсальности данного подхода, поскольку технология LLM имеет сегодня не слишком хорошую репутацию и часто выдумывает ответы («галлюцинирует»).

Рост возможностей машинного обучения, и в частности таких методов как эволюционное машинное обучение и методы символьной регрессии, приводит к смене парадигмы получения функций управления, когда современные цифровые контроллеры обучаются, то есть и структура и параметры искомых регуляторов ищутся как решение задачи оптимизации на нечисловом пространстве структур, открывая широкие возможности для получения интеллектуальных контроллеров, способных использовать нелинейности системы для повышения эффективности управления.

2.5.2. Постановка задачи численного синтеза оптимального управления

Для применения численных методов машинного обучения решения задачи синтеза системы стабилизации необходимо представить формальную математическую постановку задачи численного синтеза системы управления.

Задача численного синтеза оптимального управления состоит в нахождении структуры математического выражения многомерной функции управления и ее параметров. Размерность функции управления в общем случае равна размерности вектора управления, а количество аргументов функции управления равно размерности вектора состояния. Особенностью задачи численного синтеза управления состоит в том, что в начальный момент времени объект может находится в одном из состояний заданной области возможных начальных положений и необходимо учесть возможность нахождения объекта в любом из этих состояний.

Формально представим постановку задачи численного синтеза оптимального управления следующим образом.

Пусть задана математическая модель объекта управления в общем виде

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}), \tag{2.44}$$

где $\mathbf{x}$ – вектор состояния объекта управления, $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{u}$ – вектор управления, $\mathbf{u} \in \mathrm{U} \subseteq \mathbb{R}^m, m \leq n,$ U - компактное множество. Мы ограничиваем наш анализ всеми теми задачами управления, где U компактно, поскольку это охватывает реальные примеры задач управления, где управление ограничено, а не бесконечно, и граничные значения также принадлежат допустимому управлению.

Задана цель управления в виде некоторого терминального состояния:

$$\mathbf{x}(T) = \mathbf{x}^T \in \mathbb{R}^n, \tag{2.45}$$

где T - время достижения терминального состояния, оно может быть не задано, но ограничено $T \leq t^+$, t^+ - заданное ограниченное время достижения конечного целевого состояния.

Задана область начальных условий в пространстве состояний:

$$X_0 \subseteq \mathbb{R}^n. \tag{2.46}$$

Существование области начальных условий является основным признаком задачи численного синтеза управления. Первоначально В.Г. Болтянский [72] определил область начальных условий как целое пространство состояний $X_0 = \mathbb{R}^n$, что с прикладной точки зрения не имеет большого практического смысла. Обычно мы с некоторой погрешностью можем предположить исходную область, в которой может находиться объект и определить ее. Поэтому мы рассматриваем область X_0 как ограниченное множество в пространстве состояний.

Тогда функционал качества в общем виде можно представить следующим образом:

$$J_1 = \int_{X_0} \cdots \int \left(F\left(\mathbf{x}(T, \mathbf{x}^0)\right) + \int_0^T f_0\left(\mathbf{x}(t, \mathbf{x}^0), \mathbf{u}(t)\right) dt \right) dx_1^0 \dots dx_n^0, \qquad (2.47)$$

где $\mathbf{x}^0 = [x_1^0 \dots x_n^0]^T \in X_0$, T определяется достижением терминальных условий (2.45) и может быть разным для разных начальных условий.

Необходимо найти функцию управления как функцию вектора состояний

$$\mathbf{u} = \mathbf{h}(\mathbf{x}) \in U, \quad \mathbf{h}(\mathbf{x}): \mathbb{R}^n \to \mathbb{R}^m, \qquad (2.48)$$

такую, что при подстановке найденной функции (2.48) в модель объекта управления (2.44) получаем систему дифференциальных уравнений:

$$\dot{\mathbf{x}} = \mathbf{f}\left(\mathbf{x}, \mathbf{h}(\mathbf{x})\right), \qquad (2.49)$$

частное решение которой из любого начального состояния начальной области (2.46)

$$\mathbf{x}(0) = \mathbf{x}^0 \in X_0, \qquad (2.50)$$

достигает целевого состояния (2.45) с оптимальным значением заданного критерия качества (2.47).

Для численного решения задачи синтеза оптимального управления (2.44) – (2.48), область начальных условий (2.46) заменим конечным набором начальных условий, или сеткой начальных условий:

$$\hat{X}_0 = \{\mathbf{x}^{0,1}, \dots, \mathbf{x}^{0,K}\}. \qquad (2.51)$$

Тогда в функционале качества (2.47) интеграл по области заменится суммой по всем начальным условиям (2.51):

$$J_2 = \sum_{i=1}^{K} \left(F\left(\mathbf{x}\left(T_i, \mathbf{x}^{0,i}\right)\right) + \int_0^{T_i} f_0\left(\mathbf{x}\left(t, \mathbf{x}^{0,i}\right), \mathbf{u}(t)\right) dt \right), \quad (2.52)$$

где T_i - время достижения конечного состояния из начального состояния $\mathbf{x}^{0,i}$, $i = 1, \dots, K$.

Задача численного синтеза управления (2.44), (2.45), (2.51), (2.52) состоит в том, чтобы найти неизвестную функцию (2.48). Функция может быть задана с точностью до параметров. В общем случае, должны быть найдены как структура функции, так и ее параметры.

2.5.3. Численные методы синтеза на основе машинного обучения

Применение методов машинного обучения открывает широкие перспективы с точки зрения решения тех задач, которые раньше считались очень сложными или вообще трудно разрешимыми. Сегодня эти методы завоевывают все большую популярность у разработчиков.

Мы наблюдаем широкое применение нейросетевых технологий. Здесь хорошую эффективность показали технологии, основанные на машинном обучении с учителем, когда есть достаточный набор обучающих данных и критерием обучения является уменьшение ошибки. В работах [148, 149] представлен подход на основе аппроксимации с помощью нейронной сети множества оптимальных траекторий, рассчитанных на основе решения задачи оптимального управления из разных начальных условий. В [150] представлен тот же подход на основе аппроксимации множества программных управлений, но в качестве аппарата машинного обучения использован метод сетевого оператора. Однако, эти направления имеют вычислительные сложности с точки зрения получения достаточной обучающей выборки.

Как правило, при разработке систем управления у инженера нет необходимого набора обучающих данных, разве что получать их при ручном управлении и аппроксимировать режимы управления, произведенные оператором. Но такой подход не сильно популярен, поскольку, во-первых, скорее всего не является оптимальным, а, во-вторых, для обучения все-таки требуется значительное количество обучающих примеров управления.

К методам машинного обучения без учителя, которые показали свою эффективность для задач синтеза систем управления, относятся методы символьной регрессии [141]. Выгодная особенность этих методов состоит в том, что они позволяют искать не только оптимальные параметры функции управления, но и оптимальную структуру. Поиск структуры функции управления осуществляется на базе заданного алфавита элементарных функций. Методы символьной регрессии используют алгоритмы эволюционной оптимизации для структурно-параметрического поиска функции управления непосредственно на основе значения функционала качества. **В главе 3** диссертационного исследования подробно рассматриваются данные методы и представлены результаты, достигнутые с помощью применение данных методов в расчетах управлений для различных технических задач управления.

2.5.4. Методы расчета оптимального расположения точек равновесия в пространстве состояний.

На втором этапе принципа синтезированного оптимального управления осуществляется поиск координат точек стабилизации с помощью методов конечномерной оптимизации. В целом предлагаемый подход на основе принципа синтезированного оптимального управления выглядит как своего рода параметризация, но с существенной особенностью: параметризация выполняется после решения задачи синтеза системы стабилизации. Этот дополнительный шаг является ключевой идеей и обеспечивает достижение стабильных результатов в задачах со сложной средой или неопределенностями.

Как только дополнительное требование расширенной задачи оптимального управления выполнено за счет введения системы стабилизации объекта относительно некоторой точки равновесия в пространстве состояний, то второй этап задачи решается как задача параметрической оптимизации, где требуется найти оптимальные значения вектора параметров (2.29), минимизирующих функционал (2.28):

$$\mathbf{q}^* = \left[q_1^{*,1} \dots q_r^{*,1} \;\; q_1^{*,2} \dots q_r^{*,2} \;\dots\; q_1^{*,k} \dots q_r^{*,k}\right]^T, \qquad (2.53)$$

где r – размерность вектора параметров, влияющих на положение точки равновесия, k – количество интервалов параметризации (2.30).

Применяемый в задаче оптимального управления функционал (2.28) в общем случае не является унимодальным на пространстве искомых параметров, особенно при наличии фазовых ограничений, в связи с чем обусловлены трудности применения классических градиентных подходов.

Сегодня все большее внимание завоевывают алгоритмы глобальной оптимизации, основанные на эволюционном или популяционном поиске [48 - 50]. Эволюционные алгоритмы — это класс методов глобальной оптимизации, которые используют эвристические правила, основанные на естественных парадигмах, таких как дарвиновская эволюция, но не ограничиваясь ими, для поиска решений близких к оптимальным в сложных пространствах. Эволюционные алгоритмы не требуют дифференцируемости функции, а также способны широко исследовать пространство поиска и находить глобальный минимум с большой вероятностью.

Эволюционные вычисления имеют целый ряд выгодных особенностей для решения оптимизационных задач машинного обучения [151]:

- они не требуют вычисления градиента функции, а следовательно, не требуют дифференцируемости исследуемой функции;
- баланс между широтой и глубиной исследования пространства поиска, обеспечивающий оптимизацию в невыпуклых пространствах с многоэкстемальностью;
- возможность использовать знания по конкретной предметной области, например, ограничения могут быть введены в структуру решений, а известные решения могут быть внедрены в популяцию.
- возможность их применения для оптимизации на нечисловом пространстве кодов.

Стохастическая природа этих алгоритмов позволяет создавать новые и даже непредвиденные решения задач машинного обучения. Неожиданный

характер многих решений может быть использован для обнаружения логических недостатков в системах, разработанных людьми или другими алгоритмами.

Учитывая современные доступные мощности вычислительных процессоров, эволюционные алгоритмы сегодня с успехом внедрены и применяются в различных прикладных направлениях, даже в таких высокоточных и дорогостоящих как космические исследования [152, 153]. Например, в 2004 году Европейское космическое агентство в сотрудничестве с учеными провело серию исследований для тестирования и сравнения разных алгоритмов глобальной оптимизации для решения задач расчета межпланетных траекторий [154, 155]. Исследования показали, что эволюционные алгоритмы предлагают альтернативные и во многих отношениях превосходящие по эффективности варианты решений сложных задач глобальной оптимизации. С начала 2000-х годов различными исследователями была продемонстрирована эффективность многих эволюционных подходов, включая генетический алгоритм (Genetic Algorithm – GA) [156 - 158], оптимизационный алгоритм роя частиц (Particle Swarm Optimization – PSO) [159, 160], дифференциальная эволюция (Differential Evolution – DE) [161 - 163], оптимизационный алгоритм колонии муравьев (Ant Colony Optimization – ACO) [164, 165] и другие. Интересно, что в 2013 году адаптивный алгоритм дифференциальной эволюции, представленный на международной конференции по эволюционным вычислениям GECCO’13 был использован для планирования оптимального путешествия по спутникам Юпитера, сформированного NASA, и был награжден золотой медалью [166] за «конкурентные для человека результаты, полученные с помощью генетических и эволюционных вычислений».

В этой связи именно эволюционные алгоритмы были выбраны в качестве средств глобальной оптимизации в рамках диссертационного исследования. И хотя теоремы, представленные в [167] (no free lunch theorems), говорят о том, что ни один эволюционный подход в среднем не лучше любого другого, т.е. для любого алгоритма любое повышение производительности по одному классу задач точно оплачивается в производительности по сравнению с другим классом,

но ограничение класса оптимизационных задач задачами, представляющими интерес с точки зрения робототехники, позволяет выявить наиболее эффективные эволюционные подходы.

Далее представим некоторые эволюционные алгоритмы, которые на сегодняшний день наиболее популярны и активно применяются для параметрической оптимизации в рамках решения прикладных задач диссертационного исследования: генетический алгоритм GA [168, 169, 187 - 189], алгоритм роя частиц PSO [170 – 172], алгоритм самоорганизующейся миграции SOMA [173], модифицированный SOMA [174] и алгоритм серых волков GWO [175]. Так же в проводимых вычислительных экспериментах хорошие результаты показал разработанный гибридный алгоритм [176].

2.5.5. Эволюционные алгоритмы оптимизации параметров управления

а) Генетический алгоритм (GA – genetic algorithm)

В данной работе используется основная схема генетического алгоритма. Алгоритм включает следующие этапы.

Генерируется множество возможных решений $[\mathbf{q}^1 \dots \mathbf{q}^j]$:

$$q_i^j = \xi\left(q_i^+ - q_-^i\right) + q_i^-, i = 1, \dots, p, j = 1, \dots, H, \tag{2.54}$$

где ξ - случайное число из интервала $[0; 1]$, q_i^-, q_i^+ - нижнее и верхнее значения компоненты i параметрического вектора, $i = 1, \dots, p$, H - количество возможных решений в исходном множестве.

Для каждого возможного решения вычисляется значение критерия (2.38)

$$f_j = J(\mathbf{q}^j), j = 1, \dots, H. \tag{2.55}$$

Определяется наилучшее текущее возможное решение

$$f_{j^-} = \min\{f_1, \dots, f_H\}. \tag{2.56}$$

Первый родитель выбирается случайным образом

$$\alpha \in \{1, \dots, H\}. \tag{2.57}$$

Второй родитель β выбирается как лучший по значению функционала из случайно отобранных $D < H$

$$f_\beta = \min\{f_{j_1}, \dots, f_{j_D}\},\ j_i \in \{1, \dots, H\},\ i = 1, \dots, D. \quad (2.58)$$

Вероятность скрещивания двух особей определяется из уравнения

$$P_{GA} = \max\{\frac{f_{j^-}}{f_\alpha}, \frac{f_{j^-}}{f_\beta}\}. \quad (2.59)$$

Алгоритмически, генерируется случайное число $\xi \in [0; 1]$ и если оно меньше P_{GA}, то операция скрещивания особей выполняется.

Для выполнения этой операции необходимо определить точку скрещивания. Она определяется каждый раз случайным образом $k_c \in \{1, \dots, p\}$.

Части векторов $\mathbf{q}^\alpha$ и $\mathbf{q}^\beta$ меняются местами после точки скрещивания k_c, в результате чего получаются два новых возможных решения

$$\mathbf{q}^{H+1} = \left[q_1^\alpha \dots q_{k_c}^\alpha \ \ q_{k_c+1}^\beta \dots q_p^\beta\right]^T,\ \mathbf{q}^{H+2} = \left[q_1^\beta \dots q_{k_c}^\beta \ \ q_{k_c+1}^\alpha \dots q_p^\alpha\right]^T. \quad (2.60)$$

Далее генерируется случайное число $\xi \in [0; 1]$ и если оно не больше заданной вероятности мутации P_μ, то операция мутации по классической схеме замены одного элемента случайным возможным элементом вектора выполняется для первого нового возможного решения $\mathbf{q}^{H+1}$.

Точка мутации определяется случайным образом $k_\mu \in \{1, \dots, p\}$. Новое значение компоненты k_μ генерируется случайным образом

$$q_{k_\mu}^\alpha = \xi(q_{k_\mu}^+ - q_{k_\mu}^-) + q_{k_\mu}^-. \quad (2.61)$$

Для полученного нового возможного решения снова вычисляется значение критерия (2.28).

Находим наихудшее из возможных решений

$$f_{j^+} = \max\{f_1, \dots, f_H\}. \quad (2.62)$$

и сравниваем его с первым новым возможным решением. Если наихудшее возможное решение хуже первого нового возможного решения $f_{j^+} > f_{H+1}$, то это первое новое возможное решение вставляется во множество возможных решений вместо наихудшего возможного решения

$$\mathbf{q}^{j^+} \leftarrow \mathbf{q}^{H+1},\ f_{j^+} = f_{H+1}. \quad (2.63)$$

Те же шаги (2.62) - (2.63) выполняются для второго нового возможного решения $\mathbf{q}^{H+2}$ и если оно лучше наихудшего из возможных решений, то во

множество возможных решений также вставляется второе новое возможное решение

$$\mathbf{q}^{j^+} \leftarrow \mathbf{q}^{H+2}, \ \ f_{j^+} = f_{H+2}. \tag{2.64}$$

Этапы алгоритма (2.56) - (2.64) повторяем заданное количество раз, после чего определяется наилучшее решение, которое считается решением задачи.

б) Самоорганизующийся алгоритм миграции (SOMA – self-organizing migrating algorithm) и его модификация

Согласно алгоритму SOMA, сначала генерируется исходное множество возможных решений (2.54). Каждое решение оценивается по критерию (2.28) и создается множество оценок (2.55). Затем определяется лучшее решение (2.56).

В алгоритме заданы значения следующих параметров: s_1 – шаг перехода, $P_l > s_1$ – длина пути изменения параметров, $P_{rt} < 1$ – вероятность изменения параметра. Начальная оценка вектора эволюции задается большим числом $\tilde{f}_j = 10^{10}$. Для каждого вектора параметров $\mathbf{q}^j$, $j = 1, \dots, H$ устанавливается $t_1 = s_1$. Если $t_1 < P_l$, то генерируется случайное число $\xi \in [0; 1]$ и вычисляется новый вектор значений

$$\tilde{q}_i^j = \begin{cases} q_i^j + \left(q_i^{j^-} - q_i^j\right)t_1, & \text{если } \xi < P_{rt}, \\ q_i^j \ - \text{иначе}, \end{cases} \tag{2.65}$$

$$\tilde{q}_i^j = \begin{cases} q_i^+, \text{если } \tilde{q}_i^j > q_i^+, \\ q_i^-, \text{если } \tilde{q}_i^j < q_i^-, \end{cases} \qquad i = 1, \dots, p. \tag{2.66}$$

Компонентам нового вектора возможных решений $\tilde{\mathbf{q}}^j$ также ставим в соответствие оценки по критерию (2.28).

$$\text{Если } J\left(\tilde{\mathbf{q}}^j\right) < \tilde{f}_j, \text{ тогда } \tilde{f} \leftarrow J\left(\tilde{\mathbf{q}}^j\right). \tag{2.67}$$

Параметр t_1 увеличивается $t_1 \leftarrow t_1 + s_1$ и если $t_1 \le P_l$ то шаги (2.65)-(2.66) повторяются.

Новый полученный вектор параметров $\tilde{\mathbf{q}}^j$ сравнивается со старым вектором $\mathbf{q}^j$. Если новый вектор лучше старого, то этот старый вектор заменяется новым полученным вектором.

$$\text{Если } \tilde{f}_j, < f_j, \text{ тогда } \mathbf{q}^j \leftarrow \tilde{\mathbf{q}}^j, f_j \leftarrow \tilde{f}_j. \quad (2.68)$$

Шаги (2.65) - (2.66) выполняются заданное количество раз P. В результате решение задачи находится из множества возможных решений как наилучшее текущее решение.

Модифицированный самоорганизующийся алгоритм миграции [174] отличается дополнительными условиями при эволюции возможных решений. Лучшее решение из нескольких случайно выбранных возможных решений рассматривается вместе с лучшим текущим возможным решением для всего множества. Вместо уравнения (2.65) используется следующее уравнение

$$\tilde{q}_i^j = \begin{cases} q_i^j + v_i^j + \left(q_i^{j^-} - q_i^j\right)t_1, & \text{если } \xi < P_{rt}, \\ q_i^j \ - \text{иначе}, \end{cases} \quad (2.69)$$

где ξ - случайное число, $\xi \in [0; 1]$,

$$v_i^j \leftarrow a_1 v_i^j + a_2(q_i^{j^*} - q_i^j), \quad (2.70)$$

a_1, a_2 – параметры алгоритма, $0 \le a_1 \le 1$, $0 \le a_2 \le 1$, $\mathbf{q}^{j^*}$ – лучшее решение из w случайно отобранных возможных решений

$$f_{j^*} = \min\{f_{j_1}, \dots, f_{j_w}\}, j_k \in \{1, \dots, H\}, k = 1, \dots, w. \quad (2.71)$$

в) Алгоритм роя частиц (PSO – particle swarm optimization)

Алгоритм роя частиц в настоящее время является самым популярным популяционным алгоритмом из всех существующих.

В алгоритме заданы параметры $\alpha \in [0; 1]$, $\beta \in [0; 1]$, $\gamma \in [0; 1]$, $\sigma \in [0; 1]$.

Сначала генерируется начальное множество возможных решений (2.54). Для каждого возможного решения создается вектор скорости, первое значение которого равно нулю

$$v_i^j = 0, i = 1, \dots, p, j = 1, \dots, H. \quad (2.72)$$

Каждое возможное решение оценивается (2.55) и находится текущее лучшее решение (2.56).

Для каждого возможного решения находится наилучшее решение из нескольких случайно выбранных решений аналогично (2.71).

Значения вектора скорости пересчитываются

$$v_i^j \leftarrow \alpha\, v_i^j + \beta\xi(q_i^{j^*} - q_i^j) + \gamma\xi(q_i^{j^-} - q_i^j). \tag{2.73}$$

Каждое возможное решение преобразуется с помощью своего вектора скорости

$$\tilde{q}_i^j = q_i^j + \sigma v_i^j. \tag{2.74}$$

Затем вектор корректируется согласно ограничениям аналогично (2.66).

Новый вектор сравнивается со старым вектором по значению критерия (2.28). Если новый вектор лучше старого, то новый вектор заменяет старый.

г) Алгоритм серых волков (GWO – grey wolf optimizer)

Алгоритм серых волков отличается от других рассматриваемых алгоритмов оптимизации тем, что не имеет настраиваемых параметров.

Сначала, как и во всех популяционных алгоритмах, генерируется множество возможных решений (2.54) и его элементы оцениваются (2.55).

Далее задается цикл поколений

$$L = 2 - \frac{2k}{P}, \tag{2.75}$$

где k – номер текущего поколения, P – количество поколений.

В каждом цикле находятся k_w наилучшие возможные решения. Авторы алгоритма [175] использовали три наилучших решения, мотивируя такое количество естественным поведением стаи волков. Однако, исходя из применения данного алгоритма к различным оптимизационным задачам, целесообразно выделить количество наилучших решений в качестве настраиваемого параметра алгоритма.

$$\begin{gathered} f_{j_1} = \min\{f_j : j = 1, \dots, H\}, \\ f_{j_2} = \min\{f_j : j = 1, \dots, H\} \setminus \{f_{j_1}\}, \\ \dots \\ f_{j_{k_w}} = \min\{f_j : j = 1, \dots, H\} \setminus \{f_{j_1}, \dots, f_{j_{k_w-1}}\}. \end{gathered} \tag{2.76}$$

Для каждого возможного решения производится эволюция кроме k_w лучших.

$$\tilde{q}_i^j = \frac{1}{k_w}\sum_{r=1}^{k_w}(q_i^{j_r} - 2L(\xi-1)|2\xi q_i^{j_r} - q_i^j|), \quad (2.77)$$

где $i = 1, \dots, p, j = k_w, \dots, H$.

Проверяются ограничения на компоненты нового вектора параметров аналогично (2.66) и при необходимости корректируются. Затем оценивается новое возможное решение и, если оно лучше, то заменяется старое возможное решение на новое. После всех циклов поколений определяется наилучшее решение.

д) Гибридный эволюционный алгоритм

В ходе проведения различных экспериментов было замечено, что эволюционные алгоритмы работают быстрее, если при эволюционном преобразовании каждого возможного решения они используют как можно больше информации о значениях целевой функции в пространстве поиска. Так, например, алгоритм PSO, представленный выше в тексте диссертации, при построении новых возможных решений использует информацию о текущем наилучшем решении, о наилучшем решении среди случайно выбранных информаторов и о предыдущих значениях целевой функции для каждого возможного решения. Алгоритм GWO тоже использует информацию о некоторых текущих наилучших возможных решениях. Эти факторы обуславливают хорошую сходимость этих алгоритмов в удовлетворительную окрестность оптимального решения.

Однако, если целевая функция имеет сложный вид или включает множество сложных ограничений, то эти алгоритмы останавливаются в некоторых точках локального минимума. Генетический алгоритм в этих случаях начинает работать лучше, чем другие эволюционные алгоритмы за счет применения в качестве поискового механизма таких операций как скрещивание и мутация. GA часто может сместить поиск от текущего локального минимума.

Гибридный алгоритм, включающий в себя все три перечисленных выше алгоритма, позволяет использовать их преимущества и в практических задачах показал свою эффективность [176].

Изначально создаются все массивы и случайным образом выбирается тип эволюционного преобразования: GA, GWO или PSO. В каждом поколении количество преобразований каждого алгоритма примерно одинаково. Далее на рис. 2.4 представим псевдокод гибридного алгоритма и входящих в него алгоритмов GA (рис.2.5), PSO (рис.2.6) и GWO (рис.2.7) по отдельности.

Входные данные алгоритма: $H > 0$ – количество возможных решений в исходной популяции, $G > 0$ – количество поколений, $R > 0$ – количество эволюционных изменений в одном поколении, p – число искомых параметров, $q_i^+ > q_i > q_i^-,\ i = 1, \dots, p$ – ограничения на параметры, c – количество бит в целой части параметра, d – количество бит в дробной части параметра, α, β, γ, σ, k_0 – параметры для алгоритма PSO, а k_w – количество лидеров для алгоритма GWO.

Выходными данными алгоритма является вектор искомых параметров: $\tilde{\mathbf{q}} = \left[\tilde{q}_1 \dots \tilde{q}_p\right]^T$.

В представленном псевдокоде алгоритма функция $Goal(\mathbf{q})$ возвращает значение целевой функции для вектора параметров $\mathbf{q}$. Процедура $NumbertoGray(a, \mathbf{y})$ преобразует действительное число a в код Грея $\mathbf{y}$. Процедура $GraytoNumber(\mathbf{y}, a)$, наоборот, преобразует код Грея $\mathbf{y}$ в действительное число a. Код Грея удобен в кодировании тем, что двоичные последовательности, соответствующие двум идущим по порядку целым числам, отличаются только одним битом. Процедура $Sort(I, F, k)$ сортирует первые k элементов массива F и устанавливает первые индексы в массиве с индексом I.

Algorithm A1 Hybrid algorithm.

Require: $H > 0$ is a number of possible solutions in the initial population, $G > 0$ is the number of generations, $R > 0$ is the number of evolutionary changes in one generation, p is the number of searched parameters, $q_i^+ > q_i^-$, $i = 1, \ldots, p$, are restrictions on the parameters, c is the number bits in the integer part of the parameter, d is the number of bits in the fractional part of the parameter, α, β, γ, σ, k_0 are parameters for the PSO algorithm, and k_w is the number of leaders for the GWO algorithm.

Ensure: $\bar{\mathbf{q}} = [\bar{q}_1 \ldots \bar{q}_p]^T$ is the optimal vector of parameters

$q_j^0 = \bar{q}_j \; j = 1, \ldots, p,$
$q_j^i \leftarrow (q_j^+ - q_j^-)\xi + q_j^-, \; j = 1, \ldots, p, \; i = 1, \ldots, H-1$
$v_j^i \leftarrow 0, \; j = 1, \ldots, p, \; i = 0, \ldots, H-1$
$F_j \leftarrow Goal(\mathbf{q}^j), \; j = 0, \ldots, H-1$
$I_j = j, \; \tilde{F}_j = F_j, \; j = 0, \ldots, H-1$
$t \leftarrow 0$
while $t < G$ **do**
  $j_- \leftarrow 0, \; F_- \leftarrow F_0, \; j \leftarrow 1$
  while $j < H$ **do**
    if $F_j < F_{j_-}$ **then**
      $j_- \leftarrow j,$
      $F_{j_-} \leftarrow F_j$
    end if
    $j \leftarrow j + 1$
  end while
  $Sort(I, \tilde{F}, k_w)$
  $s \leftarrow 0$
  while $s < R$ **do**
    $k_a \leftarrow \xi(3)$
    if $k_a = 0$ **then**
      GWO transformations
    end if
    if $k_a = 1$ **then**
      GA transformation
    end if
    if $k_a = 2$ **then**
      PSO transformation
    end if
    $s \leftarrow s + 1$
  end while
  $t \leftarrow t + 1$
end while
$j_- \leftarrow 0, \; F_- \leftarrow F_0, \; j \leftarrow 1$
while $j < H$ **do**
  if $F_j < F_{j_-}$ **then**
    $j_- \leftarrow j,$
    $F_{j_-} \leftarrow F_j$
  end if
  $j \leftarrow j + 1$
end while
$\bar{\mathbf{q}} \leftarrow \mathbf{q}^{j_-}$

Рис. 2.4. Псевдокод гибридного эволюционного алгоритма

Algorithm A2 GWO transformation.

$L \leftarrow 2 - t(2/G)$
$i \leftarrow \xi(H)$
$j \leftarrow 0$
while $j < p$ **do**
  $\alpha_x \leftarrow 0$
  $k \leftarrow 0$
  while $k < k_w$ **do**
    $g_A \leftarrow 2L\xi - L$
    $g_C \leftarrow 2\xi$
    $\alpha_D \leftarrow |g_C q_j^{l_k} - q_j^i|$
    $\alpha_x \leftarrow \alpha_x + q_j^{l_k} - g_A \alpha_D$
    $k \leftarrow k + 1$
  end while
  $\hat{q}_j \leftarrow \alpha_x / k_w$
  if $\hat{q}_j > q_j^+$ **then**
    $\hat{q}_j \leftarrow q_j^+$
  end if
  if $\hat{q}_j < q_j^-$ **then**
    $\hat{q}_j \leftarrow q_j^-$
  end if
  $j \leftarrow j + 1$
end while
$\hat{F} \leftarrow Goal(\hat{\mathbf{q}})$
if $\hat{F} < F_i$ **then**
  $F_i \leftarrow \hat{F}$
  $\mathbf{q}^i \leftarrow \hat{\mathbf{q}}$
  $Sort(I, F, k_w)$
end if

Рис.2.5. Псевдокод алгоритма GWO, входящего в состав гибридного алгоритма

Algorithm A3 GA transformation.

$k_1 \leftarrow \xi(H), k_2 \leftarrow \xi(H)$
$d \leftarrow \xi$
if $d > F_{j\_}/F_{k_1}$ or $d > F_{j\_}/F_{k_2}$ **then**
  $k_s \leftarrow \xi(p)$
  $NumbertoGray(q_{k_s}^{k_1}, \mathbf{y}^1)$
  $NumbertoGray(q_{k_s}^{k_2}, \mathbf{y}^2)$
  $k_c \leftarrow \xi(c+d)$
  $j \leftarrow 0$
  while $j < k_c$ **do**
    $Sony_j^1 \leftarrow y_j^1, Sony_j^2 \leftarrow y_j^2, j \leftarrow j+1$
  end while
  $j \leftarrow k_c$
  while $j < c+d$ **do**
    $Sony_j^1 \leftarrow y_j^2, Sony_j^2 \leftarrow y_j^1, j \leftarrow j+1$
  end while
  $i \leftarrow 0$
  while $i < k_s$ **do**
    $Son_i^1 \leftarrow q_i^{k_1}, Son_i^2 \leftarrow q_i^{k_2}, i \leftarrow i+1$
  end while
  $i \leftarrow k_s + 1$
  while $i < p$ **do**
    $Son_i^1 \leftarrow q_i^{k_2}, Son_i^2 \leftarrow q_i^{k_1}, i \leftarrow i+1$
  end while
  $GraytoNumber(Sony^1, Son_{k_s}^1)$
  $GraytoNumber(Sony^2, Son_{k_s}^2)$
  $j \leftarrow 1$
  while $j \leq 2$ **do**
    if $Son_{k_s}^j > q_{k_s}^+$ **then**
      $Son_{k_s}^j \leftarrow q_{k_s}^+$
    else if $Son_{k_s}^j < q_{k_s}^-$ **then**
      $Son_{k_s}^j \leftarrow q_{k_s}^-$
    end if
    $\hat{F} \leftarrow Goal(Son^j)$
    $i_+ \leftarrow 0, i \leftarrow 1$
    while $i < H$ **do**
      if $F_i > F_{i_+}$ **then**
        $i_+ \leftarrow i$
      end if
    end while
    if $\hat{F} < F_{i_+}$ **then**
      $\mathbf{q}^{i_+} \leftarrow \mathbf{Son}^j$
      $F_{i_+} \leftarrow \hat{F}$
    end if
    $j \leftarrow j+1$
  end while
end if

Рис.2.6. Псевдокод алгоритма GA, входящего в состав гибридного алгоритма

Algorithm A4 PSO transformation.

$j \leftarrow \xi(H)$
$k \leftarrow \xi(H)$
$i \leftarrow 0$
while $i < k_0$ **do**
  $l \leftarrow \xi(H)$
  if $F_l < F_k$ **then**
    $k \leftarrow l$
  end if
end while
$i \leftarrow 0$
while $i < p$ **do**
  $v_i^j \leftarrow \alpha v_i^j + \xi\beta(q_i^k - q_i^j) + \xi\gamma(q_i^{j-} - q_i^j)$
  $\hat{q}_i \leftarrow q_i^j + \sigma v_i^j$
  if $\hat{q}_i > q_i^+$ **then**
    $\hat{q}_i \leftarrow q_i^+$
  else if $\hat{q}_i < q_i^-$ **then**
    $\hat{q}_i \leftarrow q_i^-$
  end if
  $i \leftarrow i + 1$
end while
$\hat{F} \leftarrow Goal(\hat{\mathbf{q}})$
if $\hat{F} < F_j$ **then**
  $F_j \leftarrow \hat{F}$
  $\mathbf{q}^j \leftarrow \hat{\mathbf{q}}$
end if

Рис.2.7. Псевдокод алгоритма PSO, входящего в состав гибридного алгоритма

2.6. Выводы

Рассмотрены вопросы реализации решения задачи оптимального управления. Показано, что для реализации решения необходимо сконструировать систему с обратной связью по состоянию, что чаще всего осуществляется за счет системы стабилизации. При разработке системы стабилизации движения возникают различные проблемы, например, если стабилизировать программную траекторию во времени, точное движение по оптимальной пространственной траектории не будет оптимальным движением, если оно не совпадает с ним по времени.

Предложен новый принцип синтезированного оптимального управления как способ получения реализуемого решения задачи оптимального управления. Особенность двухэтапного подхода синтезированного оптимального управления состоит в решении на первоначальном этапе задачи численного синтеза системы

управления обратной связи, позволяющей стабилизировать объект управления в некоторой точке пространства состояний. Это позволяет при дальнейшей практической реализации полученного на втором этапе оптимального управления нивелировать небольшие возмущения или неточности модели объекта управления или начального состояния.

В результате синтезированного подхода получен другой тип управления. Такое управление не является внешним воздействием на объект, управление реализовано через внутреннее состояние объекта, а точнее положение его точки равновесия в пространстве состояний.

Основная вычислительная трудность состоит в решении задачи синтеза системы стабилизации, которая в настоящее время может быть решена численно методами символьной регрессии, что будет более подробно освящено в следующей главе диссертации.

Второй этап оптимизации положения точек равновесия решается как задача параметрической оптимизации и в зависимости от условий задачи может быть решена любыми из известных методов глобальной оптимизации. В виду особенностей задач управления робототехническими системами применяемые в задачах оптимального управления функционалы в общем случае не является унимодальным на пространстве искомых параметров, особенно при наличии фазовых ограничений, в связи с чем в диссертационном исследовании предлагается использовать эволюционные алгоритмы оптимизации. Приведены наиболее популярные эволюционные алгоритмы оптимизации. Этап оптимизации параметров, влияющих на положение точки равновесия, является вычислительно менее трудоемким, чем первый этап синтеза системы стабилизации, и может быть реализован даже на бортовом процессоре робота.

Результаты, представленные в главах 2, в части обоснования и исследования свойства реализуемости моделей систем управления, были получены в рамках проекта Министерства науки и высшего образования Российской Федерации 075-15-2024-544.

Глава 3. Машинное обучение управления

Согласно представленному принципу синтезированного оптимального управления на первом этапе необходимо решить задачу синтеза системы стабилизации, обеспечивающей существование для объекта точки равновесия в пространстве состояний. Как было представлено в обзоре существующих методов, в подавляющем большинстве случаев задача синтеза управления решается аналитически или технически с учетом специфики математической модели. На сегодняшний день современные численные методы машинного обучения позволяют решать данную задачу для динамических объектов и функционалов общего вида.

Задача синтеза управления состоит в том, чтобы найти неизвестную функцию. Функция может быть задана с точностью до параметров. В общем случае, должны быть найдены как структура функции, так и ее параметры. В широком смысле машинное обучение направлено на установление некоторой функциональной зависимости, определяющей взаимосвязь между входными и выходными данными [141 – 143].

Термин машинное обучение управления был введен в книге [142]. Оперируя терминами теории оптимального управления, можно сказать, машинное обучение управления направлено на поиск закона управления некоторым объектом для оптимального достижения поставленных целей в терминах некоторого сформулированного функционала. Закон управления в общем случае представляет собой многомерную вектор-функцию, которую необходимо определить.

Сегодня машинное обучение, особенно в сфере управления, в основном используется для оптимальной настройки параметров, вроде параметров нейронной сети или какого-то заданного регулятора. В результате обучения получается оценка неизвестного вектора параметров.

Благодаря новым возможностям, которые открывают методы символьной регрессии в машинном обучении, мы теперь можем более широко определить

задачу машинного обучения управления, заключающуюся в поиске неизвестной функции управления, включая как оптимальную структуру, так и ее параметры.

Новая парадигма машинного обучения управления позволяет находить хорошие решения, близкие к оптимальным, за ограниченное время. Однако в связи с новизной этих методов возникает необходимость введения некоторых теоретических основ для обоснования результатов, полученных с помощью машинного обучения.

3.1. Теоретические основы машинного обучения управления

Во многих научных дисциплинах, если не во всех, основная задача исследований состоит в нахождении функциональной зависимости между определенными значениями параметров, характеризующих свойства исследуемого объекта. Если удается искомую функциональную зависимость представить в виде математической формулы, то очень часто такая формула становится законом в данной области, и приобретает имя его создателя. Наверное, математическая формула для искомой функциональной зависимости является венцом исследований и устанавливает факт того, что исследователь достиг определенной ступени знаний в понимании процесса и теперь он может поделиться этими знаниями в виде формулы с человечеством. Таким образом, хотелось бы подчеркнуть важность процесса поиска математического выражения для функции.

Итак, чтобы сформулировать математически задачу машинного обучения, необходимо предположить, что в исследуемом процессе существует функциональная связь между значениями некоторых параметров этого процесса. Эта функциональная связь позволяет определять значения одних параметров, называемых выходными, исходя из значений других параметров, называемых входными. В то же время математическая формула, описывающая реализацию этой функциональной связи, неизвестна. Значения входных и выходных векторов могут быть определены в результате экспериментов.

Определение 3.1. Набор вычислительных процедур, преобразующих вектор **x** из входного пространства X в вектор **y** из выходного пространства Y, и для них не существует никакого математического выражения $\mathbf{y} = \mathbf{f}(\mathbf{x})$, будем называть *неизвестной функцией*. Обозначим неизвестную функцию между входным вектором **x** и выходным вектором **y** как

$$\mathbf{y} = \alpha(\mathbf{x}). \tag{3.1}$$

Неизвестная функция (3.1) может быть представлена как некое устройство или набор экспериментальных результатов. В таком случае она будет являться черным ящиком, потому что ее точное описание неизвестно.

Определение 3.2. *Задача машинного обучения* – это задача поиска с помощью компьютерной вычислительной процедуры неизвестной функции

$$\mathbf{y} = \alpha(\mathbf{x}, \mathbf{q}), \tag{3.2}$$

где $\mathbf{y}$ – вектор значений функции, $\mathbf{y} \in \mathbb{R}^r$, $\mathbf{x}$ – вектор аргументов, $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{q}$ – вектор постоянных параметров, $\mathbf{q} \in Q \subseteq \mathbb{R}^p$, $\alpha(\mathbf{x}, \mathbf{q})$: $\mathbb{R}^n \times \mathbb{R}^p \to \mathbb{R}^r$.

Такое определение машинного обучения вполне охватывает основные задачи, которые в настоящее время решаются с помощью методов машинного обучения: аппроксимация, прогнозирование, кластеризация и т. д.

Обобщая другие различные предлагаемые в литературе определения машинного обучения [177 - 179], можно сделать вывод, что машинное обучение - это неточное численное решение некоторой математической задачи оптимизации, то есть решение, полученное машинным обучением, отличается от точного на некоторую величину, но удовлетворяет исследователя, и его можно улучшить, продолжив обучение. Во всех случаях для машинного обучения используются различные алгоритмы оптимизации, но для этих алгоритмов достаточно найти близкое к оптимальному решение.

Поиск неизвестной функции, что, как мы уже определили, является задачей машинного обучения, должен осуществляться как процесс оптимизации на основе некоторого оценочного критерия. В зависимости от типа критерия задачи машинного обучения можно разделить на два основных класса: обучение без учителя и обучение с учителем. Отметим, что существующие в настоящее

время различные виды машинного обучения можно отнести к одной из этих категорий по критерию оценки.

При обучении с учителем искомая функция аппроксимирует некоторый набор данных, который называется обучающей выборкой. Тогда функционал можно записать в следующем виде:

$$J_1 = \sum_{i=0}^{N} \left\| \hat{\mathbf{y}}^i - \alpha\left(\mathbf{x}^i, \mathbf{q}\right) \right\|, \tag{3.3}$$

где $\hat{\mathrm{Y}} = \{\hat{\mathrm{y}}^i, \dots, \hat{\mathrm{y}}^N\}$ – обучающая выборка.

При обучении без учителя эта функция используется для минимизации некоторого заданного функционала качества

$$J_2 = \int_0^T f_0(\mathbf{x}(t), \alpha(\mathbf{x}(t), \mathbf{q}))\, dt \to \min, \tag{3.4}$$

где T – время достижения цели.

Определение 3.3. *Машинное обучение* — это оптимизационная процедура поиска решения задачи в Δ-окрестности оптимального решения.

Особенность машинного обучения в том, что процедура обучения не требует достижения точного минимума критерия (3.3) или (3.4)

$$\tilde{J} \leq \min J + \Delta, \tag{3.5}$$

где Δ – некоторое положительное значение отклонения, определяющее значение функционала, достижимое при обучении. Очевидно, что для критерия (3.3) минимальное значение равно нулю. Для критерия (3.4) минимальное значение может быть неизвестно. Тогда вместо этого можно использовать предельное минимальное значение:

$$\tilde{J} = J^- + \Delta, \tag{3.6}$$

где $J^- \leq \min J$.

Исходя из введенных формулировок, методы машинного обучения можно применять в различных задачах, где требуется поиск функции. Обширной областью таких задач являются задачи управления.

Определение 3.4. Машинное обучение управления (англ. machine learning control) — это оптимизационная процедура поиска неизвестной функции управления с использованием методов машинного обучения.

К таким задачам в области управления относятся и задачи оптимального управления в различных постановках, например, в постановке Понтрягина или Беллмана, и задачи общего синтеза управления, как функции обратной связи по состоянию объекта, и задача идентификации самой модели объекта управления, как поиск функций правых частей уравнений динамики объекта. Все эти задачи требуют нахождения неизвестной функции, а значит, к ним можно применить методы машинного обучения.

При математической постановке задач управления модель объекта управления описывается системой обыкновенных дифференциальных уравнений. В правые части этой системы дифференциальных уравнений, записанной в форме Коши, входит свободный вектор управления. Этот вектор называется «свободным», потому что он может принимать любые значения из некоторого ограниченного набора значений. Без конкретных значений вектора управления невозможно решить систему обыкновенных дифференциальных уравнений, описывающую математическую модель объекта управления. Управление объектом в классическом математическом понимании заключается в качественном изменении правых частей дифференциальных уравнений за счет входящего в них вектора управления.

Когда функция управления выводится аналитически, то система гарантированно обладает требуемым свойством. В случае машинного обучения управления могут происходить события, когда система не будет обладать желаемым свойством. Назовем их плохими событиями. Например, робот достигает конечного положения почти из всех начальных условий, но не достигает его из какого-либо другого начального состояния. Хотя при хорошем обучении такие события случаются редко, они могут произойти, и вероятность их возникновения неизвестна. Нам также необходимо ввести некоторую оценку, когда мы можем считать, что вероятность плохого события мала, и можно считать обучение успешным, т. е. предположить, что система приобрела желаемое свойство.

Если в результате машинного обучения найденная функция управления должна приобрести некоторые свойства, то доказательство наличия этих свойств подтверждается моделированием и статистическим обобщением результатов моделирования.

Определение 3.5. Если проводятся D экспериментов, и в каждом i эксперименте K_i частных решений дифференциального уравнения выполняют требуемое свойство из любых $M_i \geq K_i$ случайно выбранных начальных условий из начальной области, и

$$\lim_{D\to\infty} \sum_{i=1}^{D} \frac{K_i}{M_i} \to 1, \tag{3.7}$$

существование этого свойства для дифференциального уравнения в этой области считаем *машинно обоснованным*.

Теперь мы можем переопределить некоторые свойства дифференциальных уравнений, описываемых управляемый объект, в соответствующие машинно интерпретируемые свойства.

Пусть нам дана математическая модель объекта управления. В общем случае данная модель описывается системой обыкновенных дифференциальных уравнений со свободным вектором управления в правой части:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}), \tag{3.7}$$

где $\mathbf{x}$ – вектор состояния системы, $\mathbf{u}$ – вектор управления,

$$\mathbf{x} = [x_1 \dots x_n]^T,$$
$$\mathbf{u} = [u_1 \dots u_m]^T, m \leq n,$$
$$\mathbf{f}(\mathbf{x}, \mathbf{u}) = [f_1(\mathbf{x}, \mathbf{u}) \dots f_n(\mathbf{x}, \mathbf{u})]^T.$$

В модели (3.7) многомерная функция $\mathbf{f}$ быть выражена явно на основе физических законов или идентифицирована на основе некоторых заданных функций или неявно с помощью какой-либо техники машинного обучения, например с помощью искусственных нейронных сетей [93 – 95, 180].

Задача управления, в том числе машинного обучения управления, заключается в поиске такой многомерной функции управления вместо

свободного вектора управления в правых частях системы дифференциальных уравнений:

$$\mathbf{u} = \mathbf{h}(\mathbf{x}), \tag{3.8}$$

чтобы система дифференциальных уравнений

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{h}(\mathbf{x})), \tag{3.9}$$

обладала некоторыми новыми желаемыми качественными свойствами.

Например, это могут быть такие свойства, как устойчивость решений, оптимальность и т.д.

Пусть в пространстве состояний системы дифференциальных уравнений (3.9) многообразие размерности $n - s$ определяется формулой:

$$\phi_i(\mathbf{x}) = 0, \quad i = 1, \dots, s. \tag{3.10}$$

Определение 3.6. Пусть в некоторой области $\mathrm{X} \in \mathbb{R}^n$ справедливы следующие свойства: для заданного количества K начальных условий $\mathbf{x}^{0,i} \in \mathrm{X}$, $i = 1, \dots, K$ для частного решения $\mathbf{x}(t, \mathbf{x}^{0,i})$ системы дифференциальных уравнений (3.9) $\exists\ t'$, $0 < t' \leq t^+$. Тогда

$$\left|\phi\left(\mathbf{x}\left(t', \mathbf{x}^{0,i}\right)\right)\right| \leq \Delta,\ i = 1, \dots, K, \tag{3.11}$$

где $\phi(\mathbf{x}) = [\phi_1(\mathbf{x}) \dots \phi_s(\mathbf{x})]^T$, и $\forall t\colon\ t' < t \leq t^+$

$$\left|\phi\left(\mathbf{x}\left(t, \mathbf{x}^{0,i}\right)\right)\right| < \Delta,\ i = 1, \dots, K. \tag{3.12}$$

Тогда считаем, что устойчивость системы дифференциальных уравнений (3.9) машинно обоснована на ограниченном интервале времени $(0; t^+)$ относительно многообразия (3.10).

Если размерность многообразия (3.10) равна 0, то получается машинно обоснована устойчивая точка равновесия.

Введенная машинная интерпретация существования определенных свойств объектов позволяют в дальнейшем решать сложные технические задачи машинными методами и проверять достижение требуемых свойств также на машинах.

3.2. Численные методы решения задач машинного обучения управлению

В соответствии с введенными определениями 3.1 – 3.5 методы машинного обучения ищут функцию, которая при некоторых наборах аргументов возвращает искомые значения управления, близкое к оптимальному. С одной стороны, это может снизить значимость найденного решения, но, с другой стороны, позволяет решать очень сложные задачи. Обратим также внимание на тот факт, что таких функций может быть много, и все они могут иметь различную структуру и значения параметров.

Существует два типа подходов поиска неизвестной функции: параметрический и структурно-параметрический.

- **Подход 1.** *Параметрический* подход состоит в том, что структура неизвестной функции (3.2) определяется исследователем с точностью до значений определенного числа параметров. Машинный поиск неизвестной функции в этом случае состоит в нахождении оптимальных значений параметров $\mathbf{q}$ по заданному критерию.

  К параметрическому подходу относится и случай, когда структура функции регулярно изменяется при поиске, например, если функция ищется в виде математического ряда и при поиске определяется число членов ряда.

  Поиск неизвестной функции на основе искусственных нейронных сетей также относится к параметрическому подходу. Действительно, преобразования, выполняемые в любой нейронной сети, формально описываются функцией с заданной структурой и большим количеством неизвестных параметров.
- **Подход 2.** Структурно-параметрический подход подразумевает. машинный поиск как оптимальной структуры неизвестной функции $\boldsymbol{\alpha}(\cdot)$ и оптимального значения вектора параметров $\mathbf{q}$**.**

Наиболее общим численными подходами к аппроксимации любой функции в том числе и с разрывами первого рода, является аппроксимация различными полиномиальными рядами, например, рядами Тейлора, Фурье и т.д. Но в случае многомерных функций векторного аргумента, применение таких рядов затруднительно.

Сегодня универсальным и наиболее популярным аппроксиматором считается нейронная сеть. Любая нейронная сеть представляет собой последовательный набор линейных преобразований, что в общем случае соответствует первому члену разложения в многомерный ряд Тейлора.

На первый взгляд кажется, что разнообразие видов нейронных сетей могут решить любые задачи. Однако на самом деле структура нейронной сети определяется разработчиком и представляет собой заданную структуру, в которой настраиваются только параметры, а сама структура остается неизменной. Трудно даже предположить, оптимальна ли эта структура для данной задачи. Кроме того, для сложных задач нейронная сеть имеет сложную структуру, и для инженера-разработчика, привыкшего описывать объекты и системы некоторыми функциями, имеющими физический смысл и геометрическое представление, работа с нейронной сетью представляется своего рода черным ящиком.

Сегодня структурно-параметрический подход может быть реализован с помощью методов символьной регрессии. Символьная регрессия – относительно новая технология, появившаяся позднее нейронных сетей. Она позволяет с помощью компьютера находить структуру и параметры математического выражения. В отличие от искусственных нейронных сетей, которые являются универсальным аппроксиматором любой функции, символьная регрессия также может аппроксимировать любую функцию, но получает вид этой функции в форме математического выражения. Во всех задачах, где используются нейронные сети так же могут использоваться и методы символьной регрессии. Но существуют задачи, в которых применение нейронных сетей ограничено, например, задача синтеза оптимального управления, в то время как методы

символьной регрессии показывают хорошие результаты в этой области. Все эти методы определяют базовый набор примитивных функций и правил кодирования. Затем с помощью генетического алгоритма осуществляется поиск оптимальной структуры математического выражения искомой функции на кодовом пространстве вместе с оптимальными параметрами.

Рассмотрим основные механизмы и особенности этих методов подробнее.

3.3. Символьная регрессия

Применение методов машинного обучения открывает широкие перспективы с точки зрения решения тех задач, которые раньше считались очень сложными или вообще трудно разрешимыми. Сегодня эти методы завоевывают все большую популярность у разработчиков. Наибольшим успехом пользуются технологии, основанные на машинном обучении с учителем, когда есть достаточный набор обучающих данных и критерием обучения является уменьшение ошибки [181 - 183].

Однако при разработке систем управления у инженера обычно нет необходимого набора обучающих данных, разве что получать их при ручном управлении и аппроксимировать режимы управления, произведенные оператором. Но такой подход, во-первых, скорее всего не является оптимальным, а, во-вторых, для обучения все-таки требуется значительное количество обучающих примеров управления. В итоге при обучении оптимальной системы управления разработчик может опираться только на значение функционала качества. К методам машинного обучения без учителя, которые показали свою эффективность для задач синтеза систем управления, относятся методы символьной регрессии [141 – 143, 184, 185]. Эти методы обладают целым рядом выгодных особенностей. Во-первых, они позволяют искать не только оптимальные параметры функции управления, но и оптимальную структуру, поэтому они и называются символьной регрессии, в отличие от линейной регрессии. Кроме того, символьная регрессия позволяет получать функции, в частности функции управления, в интерпретируемом для человека виде, в

отличие, например от таких методов как нейронные сети. Такая возможность методов символьной регрессии дает возможность лучше понимать механизмы управления.

Сегодня ведется много дискуссий на тему слабого и сильного искусственного интеллекта. И основная разница между ними в том, может ли научившаяся система действовать в условиях, которые не были предусмотрены в процессе обучения. Так вот особенность методов символьной регрессии в том, что они находят такие решения, которые человек и предположить не мог, поскольку осуществляют обучение только на основе значения функционала качества, т.е. заданного критерия, используя имеющийся набор элементарных блоков (алфавита элементарных функций), и поэтому результирующая функция будет действовать вне зависимости от условий, на которых ее обучали, как в случае искусственных нейронных сетей. И эта технология может рассматриваться как своего рода переход от слабого искусственного интеллекта к сильному.

Методы символьной регрессии ищут функцию управления в виде кода. Разнообразие методов символьной регрессии определяется различными способами кодирования функций. Поиск структуры функции управления осуществляется на базе заданного алфавита элементарных функций. Методы символьной регрессии используют алгоритмы эволюционной оптимизации для структурно-параметрического поиска функции управления непосредственно на основе значения функционала качества.

Методы символьной регрессии различают по способам кодирования математических выражений и совокупностью алгоритмов для поиска оптимального математического выражения на пространстве этих кодов (см. рис.3.1.).

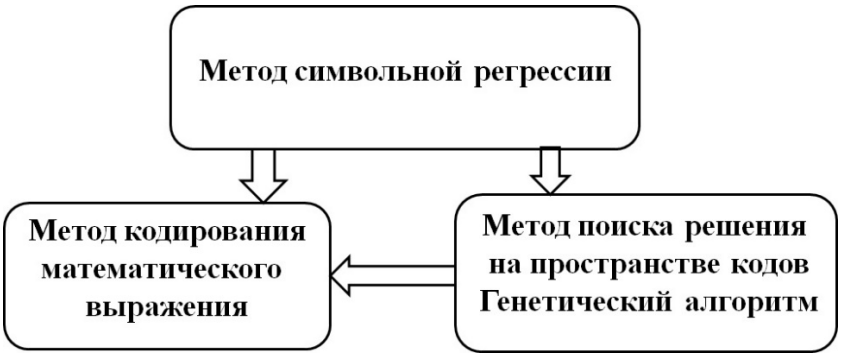

Рис.3.1. Общая структура методов символьной регрессии

3.3.1. Кодирование математических выражений

Для кодирования математического выражения методом символьной регрессии необходимо создать исходный алфавит примитивов, или базовые наборы элементарных функций. Алфавит примитивов должен включать множество элементарных функций, из которых конструируется математическое выражение и множество аргументов этого математического выражения.

Наиболее универсальным способом кодирования и удобной формой для декодирования является разбиение множества элементарных функций на подмножества функций с определенным количеством аргументов.

Возможны следующие базовые множества:

- множество функций с одним аргументом

$$\mathrm{F}_1 = \{f_{1,1}(z) = z, f_{1,2}(z), \dots, f_{1,v_1}(z)\}, \tag{3.13}$$

где $f_{1,1}(z)$ - тождественная функция, часто необходимая для кодирования выражений;

- множество функций с двумя аргументами

$$\mathrm{F}_2 = \{f_{2,1}(z_1, z_2), \dots, f_{2,v_2}(z_1, z_2)\}. \tag{3.14}$$

Все функции с двумя аргументами должны обладать следующими свойствами:

- коммутативность

$$f_{2,i}(z_1, z_2) = f_{2,i}(z_2, z_1), i = 1, \dots, v_2 \tag{3.15}$$

- ассоциативность

$$f_{2,i}\big(z_1, f(z_2, z_3)\big) = f_{2,i}\big(f_{2,i}(z_1, z_2), z_3\big), \tag{3.16}$$

- иметь единичный элемент

$$f_{2,i}(z, e_i) = f_{2,i}(e_i, z) = z, \tag{3.17}$$

где e_i – единичный элемент функции, $i = 1, \dots, v_2$.

- множество функций с тремя аргументами

$$F_3 = \{f_{3,1}(z_1, z_2, z_3), f_{3,2}(z_1, z_2, z_3), \dots, f_{3,v_3}(z_1, z_2, z_3)\}. \tag{3.18}$$

Заметим, что для описания наиболее распространенных математических выражений достаточно функций с одним и двумя аргументами. Функции с тремя и более аргументами, часто используемые, например, для описания оператора условия, тоже могут быть представлены через функции с одним и двумя аргументами, поэтому не всегда есть необходимость включать их в алфавит примитивов.

Множество аргументов математического выражения соответствует множеству функций без аргументов:

$$\mathrm{F}_0 = \{f_{0,1}, \dots, f_{0,n+p+k}\} = \{x_1, \dots, x_n, q_1, \dots, q_p, e_1, \dots, e_k\}, \tag{3.19}$$

где x_1,...,x_n – переменные, q_1,...,q_p – параметры, e_1,...,e_k – единичные элементы для функций двух аргументов.

Таким образом, алфавит примитивов можно представить в виде:

$$\begin{gathered} \mathrm{F}_0 = \{f_{0,1}, f_{0,2}, \dots, f_{\{0,v_0\}}\}. \\ \mathrm{F}_1 = \{f_{1,1}(z), f_{1,2}(z), \dots, f_{1,v_1}(z)\}, \\ \mathrm{F}_2 = \{f_{2,1}(z_1, z_2), f_{2,2}(z_1, z_2), \dots, f_{2,v_2}(z_1, z_2)\}, \end{gathered} \tag{3.20}$$

где v_i – количество элементов во множестве F_i.

Здесь индекс i при идентификаторе множества и первый индекс элемента множества указывают на количество аргументов функции.

Алфавита примитивов (3.20) достаточно для описания любого математического выражения функции управления на основании теорем Колмогорова и Арнольда о представлении (или суперпозиции) функций [186 - 188], в которых доказано, что любая непрерывная функция нескольких переменных может быть представлена в виде суперпозиций непрерывных функций одного переменного и сложения:

$$f(\mathbf{x}) = f(x_1, \dots, x_n) = \sum_{q=0}^{2n} \Phi_q \left(\sum_{p=1}^{n} \phi_{q,p}(x_p) \right),$$

где $\phi_{q,p}: [0,1] \to \mathbb{R}$ и $\Phi_q: \mathbb{R} \to \mathbb{R}$.

Представление алфавита примитивов в форме базовых множеств (3.20) не является обязательным. В некоторых методах символьной регрессии все элементарные функции не зависимо от количества аргументов размещаются в одном множестве. В любом случае при декодировании математического выражения, а это всегда требуется выполнить при вычислении значения искомого математического выражения, количество аргументов искомой функции является важной информацией.

При представлении алфавита примитивов в форме (3.20) любая элементарная функция, в том числе и аргумент математического выражения, может кодироваться целочисленным вектором из двух компонент:

$$f_{a,b} \leftrightarrow \mathbf{s} = [s_1 \; s_2]^T, \tag{3.21}$$

где s_1 – количество аргументов, s_2 – номер функции.

Далее необходимо определить правила составления математических выражений из элементарных функций. Возможны различные способы. Например, можно задать структуру функции. Затем менять в ней элементы, сохраняя первый индекс.

Наиболее универсальным способом составления математических выражений из элементарных функций является представление последовательности элементарных функций как композиции вложенных друг в друга этих элементарных функций.

Например:

$$f_{a_1,b_1} f_{a_2,b_2} f_{a_3,b_3} = f_{a_1,b_1} \circ f_{a_2,b_2} \circ f_{a_3,b_3} = f_{a_1,b_1} \left(f_{a_2,b_2} \left(f_{a_3,b_3}(\dots) \right) \right).$$

Множество элементарных функций должно обладать свойством достижимости для их использования в методах символьной регрессии.

Определение 3.6. Множество элементарных функций обладает свойствами достижимости, если для любых заданных ограниченных чисел A и B и заданной малой величины ε можно построить конечную композицию функций из этого множества

$$f_{a_1,b_1} \circ \dots \circ f_{a_k,b_k} \tag{3.22}$$

такую, что выполняется неравенство

$$\left|f_{a_1,b_1}\left(f_{a_2,b_2} \dots f_{a_R,b_K}(A) \dots\right) - B\right| \leq \varepsilon. \tag{3.23}$$

В таком случае, код математического выражения представляет собой набор кодов элементарных функций

$$\mathrm{S} = \mathbf{s}^1 \dots \mathbf{s}^L, \tag{3.24}$$

где L – длина кода функции, измеряемая количеством элементарных функций из алфавита, необходимых для описания функции, $\mathbf{s}^j = \left[s_1^j \;\; s_2^j\right]^T$,

$$s_1^j \in \{0,1,2\},$$

$$s_2^j \in \begin{cases} \{1, \dots, v_0\}, \text{если } s_1^j = 0, \\ \{1, \dots, v_1\}, \text{если } s_1^j = 1, \\ \{1, \dots, v_2\}, \quad \text{иначе.} \end{cases}$$

Очевидно, при составлении композиций простое последовательное перечисление функций может привести к нарушению математической записи. Поэтому правила составления композиций для получения правильной математической записи определены в разных методах символьной регрессии.

Приведем здесь правила составления правильных композиций из кодов функций, построенных для случая разбиения функций на подмножества (3.20) с учетом количества аргументов.

Для определения корректности записи кода, введем понятие индекса элемента композиции.

Определение 3.7. *Индекс элемента композиции* указывает на минимальное количество элементов, которое должно следовать за данным элементом.

Индекс элемента $\mathbf{s}^j = \left[s_1^j \;\; s_2^j\right]^T$ в записи композиции вычисляется по формуле

$$T(j) = T(j-1) + s_1^j - 1 \;=\; 1 - j + \sum_{i=1}^{j} s_1^i, \quad 1 \le i \le L. \tag{3.25}$$

Определение 3.8. Запись композиции является *правильной*, если все элементы записи, кроме последнего, имеют положительный индекс, а индекс последнего элемента равен нулю.

$$T(j) > 0, \quad i = 1, \dots, L-1, \tag{3.26}$$
$$T(L) = 0.$$

Теорема 3.1. *Для корректности кода математического выражения (3.24) с количеством элементов L необходимо и достаточно, чтобы выполнялись следующие соотношения*

$$1 + \sum_{i=1}^{j} s_1^i \le L, \quad j = 1, \dots, L-1, \tag{3.27}$$
$$1 - L + \sum_{i=1}^{L} s_1^i = 0. \tag{3.28}$$

Доказательство.

Рассмотрим формулу (3.27) и добавим в нее $-j$ слева и справа:

$$-j + 1 + \sum_{i=1}^{j} s_1^i \le L - j. \tag{3.29}$$

В левой части неравенства имеем:

$$T(j) = -j + 1 + \sum_{i=1}^{j} s_1^i. \tag{3.30}$$

Уравнение (3.30) вычисляет, сколько элементов из множества аргументов F_0 (3.19) должно быть после элемента j. Значение $T(j)$ увеличивается на 1 после каждого $s_1^j = 2$, не меняется после каждого $s_1^j = 1$, и убывает на 1 после каждого $s_1^j = 0$.

При $j = L$ получаем уравнение (3.28). После последнего элемента $j = L$ не должно быть элементов справа от элемента L.

Предположим, что неравенство (3.27) не выполняется. Тогда из (3.30) получаем для $j = L$

$$T(j) = -L + 1 + \sum_{i=1}^{j} s_1^i > 0.$$

Это означает, что после последнего элемента есть несколько элементов. Это обстоятельство не позволяет расшифровать код. Следовательно, условия (3.27) и (3.28) необходимы.

Пусть выполняются условия (3.27) и (3.28). Если элемент после элемента j является аргументом из множества F_0 (3.19), то $T(j)$ убывает на 1, если это номер функции с одним аргументом, то $T(j)$ не изменяется, если это номер функции с двумя аргументами, то $T(j)$ увеличивается на 1. Уравнение (3.28) показывает, что последний элемент из множества F_0 (3.19) не нуждается в аргументах. Формула расшифрована, значит выполнения условий (3.27) и (3.28) достаточно. ■.

Итак, мы рассмотрели общие принципы кодирования математических выражений, с помощью которых определяется пространство поиска в методах символьной регрессии.

3.3.2. Генетический алгоритм оптимизации на нечисловом пространстве кодов

Машинное обучение методами символьной регрессии заключается в том, что осуществляется оптимизационный поиск структуры неизвестной функции и ее параметров на пространстве кодов.

В качестве основного поискового механизма используется генетический алгоритм, который применяется как для поиска структуры математического выражения, так и параметров.

Генетический алгоритм [189 - 193] за многолетний период своего существования подтвердил свою эффективность и не нуждается в дополнительном анонсировании. Основная схема классического генетического алгоритма представлена на рис.3.2.

1. Инициализация:
- случайным образом сгенерировать множество возможных решений

2. Оценка:
- Определить значения функционала для всех возможных решений

3. Пока не выполнено условие сходимости:

3.1. Отбор:
- Выбрать «родительские решения» из множества возможных решений

3.2. Скрещивание:
- Произвольно обменивайте между собой части кодов «родительских решений» и генерируйте новые возможные решения

3.3. Мутация:
- Произвольно измените некоторые части кодов новых возможных решений

3.4. Оценка:
- Определить значения функционала для нового множества возможных решений

Рис.3.2. Общая схема классического генетического алгоритма (ГА)

Эволюционный генетический алгоритм хорошо подходит для решения оптимизационных задач символьной регрессии в виду целого ряда выгодных особенностей:

• алгоритм не требует вычисления градиента функции, а следовательно, не требует дифференцируемости исследуемой функции;

• баланс между широтой и глубиной исследования пространства поиска, обеспечивающий оптимизацию в невыпуклых пространствах с многоэкстемальностью;

• возможность использовать знания по конкретной предметной области, например, ограничения могут быть введены в структуру решений, а известные решения могут быть внедрены в популяцию.

Одной из важных особенностей генетического алгоритма является возможность его работы на пространстве кодов. Как известно классический генетический алгоритм ищет решение на числовом пространстве. Однако, поиск на пространстве кодов отличается от поиска в векторном числовом пространстве. Основная сложность его состоит в том, что метрика на пространстве кодов отличается от метрики пространства, в котором вычисляется значение целевого функционала.

Для иллюстрации указанной особенности, рассмотрим задачу нелинейного программирования:

$$f(\mathbf{q}) \to \min_{\mathbf{q}}, \tag{3.31}$$

где $\mathbf{q} \in \mathbb{R}^p$, $f(\mathbf{q})\colon \mathbb{R}^p \to \mathbb{R}^1$.

Для непрерывной целевой функции (3.31) справедливы соотношения

$$|f(\mathbf{q}^1) - f(\mathbf{q}^2)| = \delta \approx 0, \text{ если } |\mathbf{q}^1 - \mathbf{q}^2| \approx 0. \tag{3.32}$$

Пусть $G(\mathbf{q}) = \mathbf{g}$ операция перевода вещественного вектора $\mathbf{q}$ в код Грея $\mathbf{g}$, $\mathbf{g} = \left[g_1 \dots g_{\{pc\}}\right]^T$, где c – количество бит, выделенное для кодирования одной компоненты вектора $\mathbf{q}$, p – количество компонент, $g_i \in \{0,1\}$.

Введем в пространстве кодов метрику Хэмминга, которая определяет расстояние между двумя кодами, как количество несовпадающих бит

$$d_H(\mathbf{g}^1, \mathbf{g}^2) = \sum_{i=1}^{pc} \left|g_i^1 - g_i^2\right|. \tag{3.33}$$

В результате получаем, что наименьшее расстояние между кодами двух векторов равно единице.

Пусть $\mathbf{g}^1 = G(\mathbf{q}^1)$ и $\mathbf{g}^2 = G(\mathbf{q}^2)$.

Тогда, если

$$d_H(\mathbf{g}^1, \mathbf{g}^2) = 1, \tag{3.34}$$

то это не означает, что значения целевого функционала для этих векторов близки

$$f(\mathbf{q}^1) \not\approx f(\mathbf{q}^2). \tag{3.35}$$

Так происходит потому, что расчет целевого функционала выполняется с использованием компонент векторов из метрического пространства вещественных векторов.

Таким образом, задача поиска функции в закодированном виде с помощью методов машинного обучения на основе символьной регрессии является задачей нечисловой оптимизации.

Определение 3.9. Оптимизационная задача, в которой поиск решения выполняется на пространстве кодов, а вычисление целевого функционала осуществляется в метрическом векторном пространстве называется *задачей нечисловой оптимизации.*

Данная задача на сегодняшний день является одной из сложнейших в теории оптимизации. Из существующих оптимизационных методов, эволюционный генетический алгоритм способен осуществлять оптимизационный поиск на нечисловом пространстве.

Важнейшей особенностью генетического алгоритма для задач машинного обучения управления является то, что он не использует при получении новых возможных решений арифметических операций, применяя для оптимизационного поиска такие операции как скрещивание и мутация (см. рис.3.2). Эта особенность позволяет использовать его для нечисловой оптимизации.

Все методы символьной регрессии, которые применяются для машинного обучения управления, кодируют возможные решения специальным кодом и ищут оптимальное решение генетическим алгоритмом на пространстве этих кодов, при этом оценка возможного решения вычисляется в пространстве вещественных функций.

Основная трудность, возникающая при реализации процесса поиска решения в задаче нечисловой оптимизации, является сложность пространства поиска. Поиск организован на нечисловом пространстве кодов функций, где может быть задана лишь некоторая символьная метрика, например расстояние Левенштейна, Хемминга или Яро. Однако оценка решений при поиске производится в пространстве функций с совершенно другой метрикой. Получается, что процесс поиска осуществляется на пространстве кодов функций, где нет единой метрики. Как и в пространстве слов: есть алфавит и слова могут быть близки, исходя из оценки символов, но иметь совершенно разные значения, исходя из семантической оценки. Близость слов не соответствует близости значений. То же самое и с поиском в пространстве кодов функций. Оценка функции работает с отображениями. Поиск осуществляется по кодам. Таким образом, метрика между именами функций не соответствует расстояниям между значениями функций. Вероятно, этим можно объяснить тот факт, что методы символьной регрессии, несмотря на широкий спектр своих возможностей, еще

не сформировались как мощный инструмент в решении задачи машинного обучения управлению.

Сложностью пространства поиска определяется необходимость введения дополнительных механизмов, облегчающих и ускоряющих поиск оптимальных решений на нечисловом пространстве кодов. Одним из таких механизмов предлагается внедрение принципа малых вариаций базисного решения.

3.4. Принцип малых вариаций базисного решения

Во всех методах символьной регрессии основным поисковым механизмом являются эволюционные алгоритмы. Большинство известных эволюционных алгоритмов включают арифметические операции для преобразования возможных решений и получения эволюции. Однако, поиск оптимального решения в пространстве кодов сложен тем, что это пространство не имеет единой метрики. Для таких пространств поиска нельзя использовать ни классические градиентные алгоритмы, ни эволюционные алгоритмы с арифметическими операциями. Таким образом, генетический алгоритм является основным алгоритмом поиска в пространстве кодов, не использующим в своих этапах арифметические операции. Иногда при определенных сложных формах кодирования построение новых возможных решений с помощью операций генетического алгоритма, скрещивания и мутации, является существенной проблемой.

Исследования этой проблемы привели к формулировке принципа малых вариаций базисного решения. Этот принцип был впервые предложен Дивеевым А.И. [194, 195] и применен в рамках разработанного им метода символьной регрессии, метода сетевого оператора.

Суть этого подхода состоит в том, что поиск математического выражения можно начинать в окрестности одного заданного возможного решения. Это решение кодируется методом символьной регрессии и называется базисным решением. В сложных задачах целесообразно использовать такое базисное

решение для ускорения процесса поиска. Для базисного решения определяется множество возможных малых вариаций. Малая вариация – это такое незначительное изменение в коде, которое приводит к появлению нового возможного решения. Малые вариации изменяют код так, что в результате получается правильный код нового возможного решения. Согласно этому принципу, новые возможные решения кодируются как наборы малых вариаций базисного решения. Чтобы получить код нового возможного решения, необходимо применить к базисному решению вектор малых вариаций. Генетические операции применяются к вектору вариаций. Через несколько поколений базисное решение можно заменить на лучшее текущее решение.

Такой подход на основе принципа малых вариаций базисного решения очень удобен для поиска систем управления, т.к. специалист в управлении интуитивно или на основе своего опыта всегда может предположить некоторую хорошую рабочую систему управления. Эту систему управления можно рассматривать как базисное решение.

Рассмотрим подробно принцип малых вариаций базисного решения.

Пусть имеется пространство кодов Ξ^n. Любой элемент этого пространства представляет собой вектор кода

$$\mathbf{y} = [y_1 \dots y_n]^T \tag{3.36}$$

некоторой нечисловой конструкции, в частности математического выражения. Любой элемент этого кода имеет значение из множества возможных символов

$$y_i \in A = \{0, a_1, \dots, a_L\}. \tag{3.37}$$

В некоторых случаях символом для кода элемента может быть целое число. Тогда кодом нечислового элемента будет набор целых чисел, для которых неприменимы арифметические операции, как, например, для почтовых кодов.

Определение 3.10. *Элементарной вариацией* кода нечислового элемента является замена значения одного элемента кода на другое значение из множества возможных символов.

Не всегда замена одного символа на другой может привести к новому правильному коду, который соответствует некоторой другой нечисловой конструкции.

Определение 3.11. *Малой вариацией* кода нечисловой конструкции является минимальный набор элементарных вариаций, который позволяет получить новый допустимый код нечисловой конструкции.

Для заданного пространства кодов определим множество малых вариаций

$$\Omega(\Xi^n) = \{\delta_1(\mathbf{y}), \dots, \delta_M(\mathbf{y})\}. \quad (3.38)$$

Малых вариаций в зависимости от кода может быть несколько.

Определение 3.12. Набор малых вариаций является полным, если он позволяет получить из любого допустимого кода нечисловой конструкции любой другой допустимый код из пространства кодов.

Введем понятие расстояния на пространстве кодов.

Определение 3.13. *Расстояние между двумя элементами* $\mathbf{y}^1$, $\mathbf{y}^2$ пространства кодов Ξ^n – это минимальное количество малых вариаций одного кода $\mathbf{y}^1$, требуемых для получения другого кода $\mathbf{y}^2$

$$\|\mathbf{y}^1 - \mathbf{y}^2\|_\Xi = d, \quad (3.39)$$

где

$$d = \min_r\{\mathbf{y}^2 = \delta_{k_1}(\dots \delta_{k_r}(\mathbf{y}^1) \dots)\}. \quad (3.40)$$

Определение 3.14. Δ-окрестностью кода $\Delta(\tilde{\mathbf{y}})$ является подмножество всех кодов, которые находятся от кода $\tilde{\mathbf{y}}$ на расстоянии не более, чем Δ

$$\forall \mathbf{y} \in \Delta(\tilde{\mathbf{y}}) \Rightarrow \|\tilde{\mathbf{y}} - \mathbf{y}\|_\Xi \leq \Delta. \quad (3.41)$$

Для кодирования малой вариации введем целочисленный вектор

$$\mathbf{w} = [w_1 \dots w_r]^T, \quad (3.42)$$

где r – размерность вектора вариаций, которая зависит от формы кодирования конкретного метода символьной регрессии, w_1 – номер или тип малой вариации, остальные компоненты вектора вариаций определяют номер варьируемого элемента в коде, эти компоненты включают информацию о местоположении изменяемого элемента и его новом значении.

Например, чтобы записать небольшую вариацию в декартовом генетическом программировании, достаточно использовать целочисленный вектор с тремя компонентами.

$$\boldsymbol{w} = [w_1\ w_2\ w_3]^T, \tag{3.43}$$

где w_1 – номер столбца в коде, w_2 – номер строки в столбце w_1, w_3 – новое значение элемента.

Вектор малой вариации метода сетевого оператора состоит из четырех элементов

$$\boldsymbol{w} = [w_1\ w_2\ w_3\ w_4]^T \tag{3.44}$$

где w_1 - тип вариации, w_2 - номер строки, w_3 - номер столбца матрицы оператора сети, а w_4 - новое значение элемента.

Возможные решения закодированы как множества векторов малых вариаций, применяемые к базисному решению:

$$\mathrm{W}^i = \{\boldsymbol{w}^{i,1}, \dots, \boldsymbol{w}^{i,d}\}, \tag{3.45}$$

где d - длина или глубина вариаций.

Генетический алгоритм выполняет эволюцию на множествах векторов малых вариаций. Рассмотрим подробнее этапы генетического алгоритма поиска оптимального решения на основе принципа малых вариаций базисного решения.

1. Устанавливаем базисное решение, которое по мнению исследователя является наиболее близким к оптимальному возможному решению.

$$\mathbf{y}^0 = [y_1^0 \dots y_n^0]^T. \tag{3.46}$$

В практических задачах синтеза управления в качестве примера базисного решения для различных объектов управления можно использовать пропорциональный регулятор или линейное преобразование вектора состояния.

2. Генерируем множество возможных решений в форме упорядоченных мультимножеств векторов вариаций. Количество элементов в этих наборах определяет окрестность базисного решения

$$\mathrm{W}^i = \{\boldsymbol{w}^{i,1}, \dots, \boldsymbol{w}^{i,d}\},\ i = 1, \dots, H, \tag{3.47}$$

где H – количество возможных решений в начальной популяции, d – количество векторов вариаций в одном наборе.

Каждое возможное решение в начальной популяции получается в результате малых вариаций базисного решения

$$\mathbf{y}^i = \boldsymbol{w}^{i,d} \circ \dots \circ \boldsymbol{w}^{i,1} \circ \mathbf{y}^0. \tag{3.48}$$

Из определения 3.14 и (3.48) следует, что каждое возможное решение в популяции принадлежит d -окрестности базисного решения.

$$\mathbf{y}^i \in d(\mathbf{y}^0),\ i = 1, \dots, H. \tag{3.49}$$

3. Вычисляем значение целевого функционала для каждого возможного решения в популяции

$$F_i = J\left(\Omega\left(\mathbf{y}^i\right)\right),\ i = 1, \dots, H, \tag{3.50}$$

где $\Omega\left(\mathbf{y}^i\right)$ – функция преобразования кода нечисловой конструкции в вещественную функцию.

4. Пока условие останова не выполнено, реализуется цикл эволюции.

   a. Случайно отбираем два набора векторов вариаций

$$\begin{aligned} \mathrm{W}^\alpha &= \{\boldsymbol{w}^{\alpha,1}, \dots, \boldsymbol{w}^{\alpha,d}\}, \\ \mathrm{W}^\beta &= \{\boldsymbol{w}^{\beta,1}, \dots, \boldsymbol{w}^{\beta,d}\}. \end{aligned} \tag{3.51}$$

   b. Вычисляем вероятность скрещивания по значениям целевого функционала для отобранных векторов

$$Pr_c = \max\left\{\frac{F_{j\_}}{F_\alpha}, \frac{F_{j\_}}{F_\beta}\right\}. \tag{3.52}$$

   c. Если генератор случайных чисел выдал число меньше, чем Pr_c, то выполняем операцию скрещивания.

   Находим случайно точку скрещивания

$$k_c \in \{1, \dots, d\}. \tag{3.53}$$

Обмениваем вектора вариаций в отобранных наборах после точки скрещивания и получаем два новых набора векторов вариаций, которые соответствуют двум новым решениям из d-окрестности базисного решения

$$\begin{aligned} \mathrm{W}^{\gamma} &= \{\boldsymbol{w}^{\alpha,1}, \dots, \boldsymbol{w}^{\alpha,k_c}, \boldsymbol{w}^{\beta,k_c+1}, \dots, \boldsymbol{w}^{\beta,d}\}, \\ \mathrm{W}^{\delta} &= \{\boldsymbol{w}^{\beta,1}, \dots, \boldsymbol{w}^{\beta,k_c}, \boldsymbol{w}^{\alpha,k_c+1}, \dots, \boldsymbol{w}^{\alpha,d}\}. \end{aligned} \tag{3.54}$$

d. Далее для полученных новых возможных решений в виде наборов векторов вариаций (3.54) выполняем операцию мутации с заданной вероятностью. Для этого для каждого нового возможного решения находим случайно точку мутации и генерируем новый вектор вариаций в этой позиции.

e. Затем вычисляем значения целевого функционала для полученных новых возможных решений и определяем по значениям этих функционалов судьбу каждого нового возможного решения, либо оно отбрасывается, либо включается в популяцию вместо наиболее плохого в данный момент возможного решения.

Таким образом, генетический алгоритм на основе принципа малых вариаций базисного решения включает те же действия, что и обычный генетический алгоритм, и операция скрещивания в нем выполняется обычным способом, обменом хвостовых элементов после точки скрещивания. Единственным отличием этого алгоритма от обычного генетического алгоритма является смена базисного решения после выполнения определенного количества операций скрещивания и мутации. Для этого находится наилучшее найденное к этому моменту возможное решение. Оно кодируется методом символьной регрессии и устанавливается вместо базисного решения.

Такой подход может показаться двойным кодированием возможных решений, но он дает два преимущества. Во-первых, использование базисного решения дает ориентир для поиска в сложном пространстве функций и значительно ускоряет поиск. Во-вторых, применение операций генетического алгоритма не к кодам возможных решений непосредственно, а к векторам

вариаций позволяет всегда получать правильные коды возможных решений без необходимости введения дополнительных проверок.

Таким образом, принцип малых вариаций может быть применен к любому известному методу символьной регрессии для преодоления трудностей, связанных с решением задачи синтеза управления. Принцип малых вариаций базисного решения впервые был использован в методе сетевого оператора. Далее этот принцип применялся в других методах символьной регрессии. В следующих разделах будут представлены разработанные вариационные методы символьной регрессии на основе применения принципа малых вариаций базисного решения. Во всех случаях его применения метод символьной регрессии при поиске математического выражения работал существенно лучше, чем тот же метод символьной регрессии без принципа малых вариаций базисного решения.

3.5. Генетический алгоритм многокритериального структурно параметрического поиска функций управления

Особенность задач синтеза управления состоит в том, что при поиске функции управления важно вместе со структурой функции управления искать вектор параметров, который входит в функцию управления как часть из аргументов. Введение параметров в структуру функции является естественным расширением класса искомых функций.

Заметим, что изначально в методах символьной регрессии, и в частности, в самом известном и популярном методе генетического программирования, параметры вводятся как константы, входящие в искомую функцию в качестве аргументов, а дальнейшее их изменение осуществляется за счет преобразования с помощью элементарных функций. Такой подход не совсем логичен. Предположим, получена приемлемая структура функции управления. Единственным недостатком этой структуры является то, что параметры, входящие в нее, не являются оптимальными. Алгоритм продолжит изменять структуру, чтобы изменить параметры, т.е. алгоритм будет усложнять структуру, заменяя параметры различными функциями этих параметров. Например, если

параметр имеет значение 5, а оптимальное значение этого параметра для заданной структуры функции 1,57, то трудно представить, сколько нелинейных преобразований параметра надо сделать, чтобы получить значение 1,57 из значения 5. Кроме того, все эти нелинейные преобразования затем должны быть включены в реализуемую функцию управления.

Напрашивается разумный вывод, что значения параметров имеет смысл искать вместе со структурой функции, в которую они входят. Поиск структуры и параметров должен осуществляться в рамках одного и того же генетического алгоритма, но на разных структурах данных. Именно такой подход представлен в диссертационном исследовании.

Еще одной особенностью численного решения задач управления является, как правило, наличие нескольких критериев качества. В задачах управления всегда есть цель управления, которая часто формулируется в виде терминальных условий, и критерий качества управления, который формулируется в виде интегрального функционала. При численном синтезе необходимо учитывать точность достижения цели и ее влияние на оценку критерия качества. На практике всегда можно свернуть критерии с определенными весами, но для генетического алгоритма такая свертка не дает большого преимущества. Поэтому в диссертации предложен поиск решений в виде множества оптимальных по Парето решений.

Представим описание многокритериального генетического алгоритма структурно-параметрического поиска функции.

1. Введем исходные данные для алгоритма поиска.

Пусть задан векторный критерий качества

$$\mathbf{J}(\mathrm{S}, \mathbf{q}) = [j_1(\mathrm{S}, \mathbf{q}) \ldots j_v(\mathrm{S}, \mathbf{q})]^T, \qquad (3.55)$$

где S - код структуры искомой функции, $\mathbf{q}$ - вектор параметров.

Задан код базисного решения и значение вектора параметров для этого решения

$$\mathrm{S}_0,\ \mathbf{q}^0 = \left[\overbrace{\{1 \dots 1\}}^{p}\right]^T. \tag{3.56}$$

Определены параметры алгоритма: H - количество возможных решений в популяции, P - число поколений, R - число возможных скрещиваний в одном поколении, α - минимальное значение вероятности скрещивания, E - число поколений в одной эпохе или число поколений между сменой базисного решения, p_μ - вероятность выполнения операции мутации, c - число бит под целую часть параметра, d - число бит под дробную часть параметра, M - число малых вариаций для одного возможного решения, p - размерность вектора параметров.

2. Генерируем множество возможных решений в виде упорядоченных множеств векторов малых вариаций

$$\mathrm{W} = \{\mathrm{W}_1, \dots, \mathrm{W}_{H\}}, \tag{3.57}$$

$$\mathrm{W}_i = \left(\mathbf{w}^{i,1}, \dots, \mathbf{w}^{i,M}\right),\ i = 1, \dots, H.$$

Задаем множество нулевых вариаций, которые не варьируют базисное решение

$$\mathrm{W}_0 = (\mathbf{w}^{0,1}, \dots, \mathbf{w}^{0,M}), \tag{3.58}$$

$$\mathbf{w}^{0,i} = [0 \dots 0]^T,\ \ i = 1, \dots, M.$$

Данный нулевой вектор вариаций не меняет кода базисного решения

$$[0 \dots 0]^T \circ \mathrm{S}_0 = \mathrm{S}_0.$$

Множество нулевых векторов вариаций введены для того, чтобы базисное решение также участвовало в операциях скрещивания.

Генерируем множество двоичных кодов для векторов параметров возможных решений

$$\mathrm{Z} = \{\mathbf{z}^1, \dots, \mathbf{z}^H\}, \tag{3.59}$$

$$\mathbf{z}^i = \left[z_1^i \dots z_{p(c+d)}^i\right]^T,$$

$$z_j^i = \xi(2),$$

где $\xi(A)$ - функция генератора случайных чисел, при каждом обращении к ней, она возвращает случайное целое число от 0 до $A-1$.

Преобразуем вектор параметров базисного решения в код Грея

$$\mathbf{z}^0 = Gray(\mathbf{q}^0), \tag{3.60}$$

где $Gray(\mathbf{q})$ - функция преобразования вещественного вектора в код Грея.

3. Вычисляем значение функционалов для всех возможных решений, включая базисное:

$$\begin{gathered} \mathrm{S}_i = \mathbf{w}^{i,M} \circ \ldots \circ \mathbf{w}^{i,1} \circ \mathrm{S}_0, \\ \mathbf{q}^i = Gray^{-1}(\mathbf{z}^i), \\ \mathbf{f}^i = \mathbf{J}(\mathrm{S}_i, \mathbf{q}^i), \;\; i = 0, \ldots, H, \end{gathered} \tag{3.61}$$

где $Gray^{-1}(\mathbf{z})$ – функция обратного перевода из кода Грея в вещественный вектор.

Вычисляем значения рангов Парето для всех значений функционалов

$$\begin{gathered} L_i = 0, \\ L_i \leftarrow L_i + 1, \text{если } \mathbf{f}^j < \mathbf{f}^i, \;\; j = 0, \ldots, H \\ i = 0, \ldots, H, \end{gathered} \tag{3.62}$$

где

$$\mathbf{f}^j < \mathbf{f}^i, \text{ если } f_j^k \le f_k^i, \;\; k = 1, \ldots, v \;\; \text{и} \;\; \exists r, \;\; 1 \le r \le v \;\; \Rightarrow f_r^j < f_r^i.$$

Множество Парето определяется элементами с нулевыми значениями ранга Парето

$$Pareto = \{i_1, \ldots, i_K\}, \;\; L_{i_j} = 0, \;\; j = 1, \ldots, K. \tag{3.63}$$

4. Запускаем оптимизационный поиск - цикл эволюции.

Шаг 1. Начало цикла поколений. Устанавливаем счетчик поколений $j_p = 0$ и запускаем цикл по поколениям.

Шаг 2. Начало цикла скрещиваний. Устанавливаем счетчик количества возможных скрещиваний $j_c = 0$.

Отбираем возможные решения для скрещивания

$$a = \xi(H), \;\; b = \xi(H). \tag{3.64}$$

Вычисляем вероятность скрещивания

$$Pr_c = \max\left\{\frac{1+\alpha L_a}{1+L_a}, \frac{1+\alpha L_b}{1+L_b}\right\}. \quad (3.65)$$

Шаг 3. Проверяем выполнение условий

$$\xi \le Pr, \quad (3.66)$$

где ξ – функция генератора случайных чисел, при каждом обращении к ней, она возвращает случайное число из интервала $(0; 1)$.

Если условие (3.66) не выполняется, то переходим на шаг 9.

Заметим, что, согласно формуле (3.66), вероятность скрещивания равна единице, если один из родителей принадлежит множеству Парето и имеет ранг Парето равный 0.

Шаг 4. Выполняем операцию скрещивания. Определяем случайно точки скрещивания для структурной и параметрической частей

$$r = \xi(M), \;\; s = \xi\big(p(c+d)\big). \quad (3.67)$$

Выполняем скрещивание, получаем четыре новых возможных решения. Два новых возможных решения получены путем скрещивания и структурной и параметрической частей, два других новых возможных решения получены путем скрещивания только параметрических частей

$$\begin{gathered}
\mathrm{W}_{H+1} = (\mathbf{w}^{a,1}, \dots, \mathbf{w}^{a,r-1}, \mathbf{w}^{b,r}, \dots, \mathbf{w}^{b,M}),\\
\mathrm{W}_{H+2} = (\mathbf{w}^{b,1}, \dots, \mathbf{w}^{b,r-1}, \mathbf{w}^{a,r}, \dots, \mathbf{w}^{a,M}),\\
\mathrm{W}_{H+3} = \mathrm{W}_a,\\
\mathrm{W}_{H+4} = \mathrm{W}_b,\\
\mathbf{z}^{H+1} = \left[z_1^a \dots z_{s-1}^a \; z_s^b \dots z_{p(c+d)}^b\right]^T,\\
\mathbf{z}^{H+2} = \left[z_1^b \dots z_{s-1}^b \; z_s^a \dots z_{p(c+d)}^a\right]^T,\\
\mathbf{z}^{H+3} = \left[z_1^a \dots z_{s-1}^a \; z_s^b \dots z_{p(c+d)}^b\right]^T,\\
\mathbf{z}^{H+4} = \left[z_1^b \dots z_{s-1}^b \; z_s^a \dots z_{p(c+d)}^a\right]^T.
\end{gathered} \quad (3.68)$$

Шаг 5. Выполняем операцию мутации для каждого нового возможного решения $H + i, i = 1, \dots, 4$.

Проверяем условия выполнения мутации

$$\xi \le p_\mu. \quad (3.69)$$

Если условие (3.69) выполняется, то операции мутации выполняется. Случайно определяем позиции в структурной и параметрической частях

$$m_i = \xi(M),\ \ l_i = \xi\big(p(c+d)\big), \tag{3.70}$$

и генерируем новый вектор вариации и элемент в параметрической части

$$\mathbf{w}^{H+i,m_i},\quad z_{l_i}^{H+i} = \xi(2). \tag{3.71}$$

Шаг 6. Вычисляем значения функционалов для каждого нового возможного решения

$$\mathbf{f}^{H+i} = \mathbf{J}\big(\mathrm{S}_{H+i}, \mathbf{q}^{H+i}\big), \tag{3.72}$$

где

$$\mathrm{S}_{H+i} = \mathbf{w}^{H+i,M} \circ \ldots \circ \mathbf{w}^{H+i,1} \circ \mathrm{S}_0,$$
$$\mathbf{q}^{H+i} = Gray^{-1}\big(\mathbf{z}^{H+i}\big),\ i = 1, \ldots, 4.$$

Шаг 7. Вычисляем значение ранга Парето для новых возможный решений L_{H+i}.

Определяем возможное решение с наибольшим значением ранга Парето

$$L_{j+} = \max\{L_k : k = 1, \ldots, H\}. \tag{3.73}$$

Шаг 8. Проверяем условие замены решений

$$L_{H+i} < L_{j+}. \tag{3.74}$$

Если условие (3.74) выполняется, т.е. новое возможное решение $H+i$ имеет величину ранга Парето меньше, чем наибольшее значение ранга Парето для всего множества возможных решений, то заменяем возможное решение с наибольшим значением ранга Парето, на новое полученное возможное решение

$$\mathrm{W}_{j+} \leftarrow \mathrm{W}_{H+i}, \tag{3.75}$$
$$\mathbf{z}^{j+} \leftarrow \mathbf{z}^{H+i},$$
$$L_{j+} \leftarrow L_{H+i},$$
$$\mathbf{f}^{j+} \leftarrow \mathbf{f}^{H+i}.$$

Шаги 7, 8 повторяем для всех новых возможных решений $i = 1, \ldots, 4$.

Шаг 9. Увеличиваем счетчик числа скрещиваний

$$j_c \leftarrow j_c + 1. \tag{3.76}$$

Шаг 10. Проверяем условия выхода из цикла скрещиваний.

Если $j_c < R$, то переходим на шаг 2.

Шаг 11. Увеличиваем счетчик числа поколений

$$j_p \leftarrow j_p + 1. \tag{3.77}$$

Шаг 12. Проверяем условие конца эпохи, или смены базисного решения.

$$j_p \bmod E = 0. \tag{3.78}$$

Если условие (3.78) выполняется, то заменяем базисное решение на новое лучшее найденное решение, иначе переходим на шаг 15.

Шаг 13. Меняем базисное решение.

Для всех возможных решений, включая базисное нормируем значения функционалов

$$\tilde{f}_i^{\,j} = \frac{f_i^j - f_j^{j-}}{f_i^{j+} - f_i^{j-1}},\ i = 1, \dots, v, ;\ j = 0, \dots, H, \tag{3.79}$$

где

$$f_i^{j-} = \min\{f_i^j : j = 0, \dots, H\},\ \ 1 \le i \le v,$$

$$f_i^{j+} = \max\{f_i^j : j = 0, \dots, H\},\ \ 1 \le i \le v.$$

Вычисляем нормы функционалов для всех возможных решений

$$\tilde{f}_j = \sqrt{\sum_{i=1}^{v} \left(\tilde{f}_i^{\,j}\right)^2},\ j = 0, \dots, H. \tag{3.80}$$

Определяем новое базисное решение, имеющее наименьшее значение нормы функционала

$$\tilde{f}_{j_0} = \min\{\tilde{f}_j :\ j = 0, \dots, H\} \tag{3.81}$$

Меняем базисное решение.

$$\mathrm{S}_{j_0} = \mathbf{w}^{j_0, M} \circ \dots \circ \mathbf{w}^{j_0, 1} \circ \mathrm{S}_0, \tag{3.82}$$

$$\mathrm{S}_0 \leftarrow \mathrm{S}_{j_0},$$

$$\mathbf{z}^0 \leftarrow \mathbf{z}^{j_0},$$

$$\mathbf{f}^0 \leftarrow \mathbf{f}^{j_0}.$$

Шаг 14. Вычисляем значения функционалов для всех возможных решений по формулам (3.61) и значения рангов Парето для всех возможных решений по формулам (3.62).

Шаг 15. Проверяем условие окончания вычислений

$$j_p \geq P. \tag{3.83}$$

Если условие (3.83) не выполнено, то переходим на шаг 1, иначе завершаем вычисления.

Решением задачи является множество Парето оптимальных решений, которое определяем по нулевым значениям ранга Парето. В дальнейшем исследователю предлагается выбрать в качестве решения задачи одно из возможных решений на множестве Парето. Если исследователя не удовлетворяет ни одно из решений на множестве Парето, то он выбирает одно решение, которое определяет как базисное и запускает алгоритм сначала.

3.6. Пространство машинно реализуемых функций

Машинный поиск функции имеет такую особенность, что в процессе поиска функции могут принимать бесконечные значения, что приводит к остановке процесса поиска в виде сообщения об ошибке переполнения. Для того, чтобы избежать возникновения этого события, необходимо изменить процесс вычисления элементарных функций из алфавита так, чтобы они не принимали недопустимых значений. Более того, необходимо обеспечить удовлетворение требования о том, что найденные функции управления будут реализованы на бортовом процессоре робота или любом другом автоматическом устройстве. Это означает, что найденные функции управления должны быть реализуемы компьютером и их модульное значение никогда не будет принимать значение бесконечности.

Этим обусловлена необходимость ввести некоторое пространство, отличное от $\mathbb{R}^n$, чтобы исключить ошибки переполнения.

Рассмотрим такое пространство функций.

Это пространство является подпространством вещественного векторного пространства

$$\mathbb{R}_{\#}^{n} \subseteq \mathbb{R}^{n}, \tag{3.84}$$

где $\mathbb{R}_{\#}^{n}$ – машинно реализуемое пространство.

Это пространство $\mathbb{R}_{\#}^{n}$ обладает следующими свойствами. Для любого вектора $\mathbf{x} = [x_1 \dots x_n]^T \in \mathbb{R}_{\#}^{n}$ размерности n выполняются следующие условия:

1)

$$|x_i| \le B^+ < \infty,\ i = 1, \dots, n. \tag{3.85}$$

2) существует малое положительное значение $\delta^- > 0$, что

$$\text{если } |x_i| < \delta_i^-,\ \text{ то } x = 0,\ i = 1, \dots, n. \tag{3.86}$$

3)

$$\text{если } \mathbf{x}(t) \in \mathbb{R}_{\#}^{n},\ \text{ то } \dot{\mathbf{x}}(t) \in \mathbb{R}_{\#}^{n}. \tag{3.87}$$

4) существует значение удовлетворительной точности $\widetilde{\Delta} > \delta^-$, что

$$\text{если } |\alpha| < \widetilde{\Delta},\ \text{ то } x_i \pm \alpha = x_i,\ \ i = 1, \dots, n. \tag{3.88}$$

Обычно в задачах с дифференциальными уравнениями значение удовлетворительной точности составляет полшага интегрирования. Производная функции в машинно реализуемом пространстве $\mathbb{R}_{\#}^{n}$ вычисляется по соотношению

$$\frac{\partial f(z)}{\partial z} = \frac{f(z+\delta^-) - f(z)}{\delta^-}. \tag{3.89}$$

Рассмотрим пример.

$$\frac{\partial \sin(z)}{\partial z} = \frac{\sin(z+\delta^-) - \sin(z)}{\delta^-} =$$

$$\frac{\sin(z)cos(\delta^-) + \sin(\delta^-)\cos(z) - \sin(z)}{\delta^-} = \langle \text{используя}(*) \rangle =$$

$$\frac{\sin(z) + \delta^- \cos(z) - \sin(z)}{\delta^-} = \cos(z).$$

Для преобразований использовалось следующее равенство

$$\cos(\delta^-) = 1 - 0.5(\delta^-)^2 = 1,$$

согласно (3.88).

Во введенном пространстве $\mathbb{R}_{\#}^{n}$ машинно реализуемые функции записываются как обычные функции из математического анализа, но с условием, что их значения никогда не равны бесконечности.

Например,

$$z^{-1} = \begin{cases} \frac{1}{z}, \text{если } |z| > \delta^{-}, \\ \operatorname{sgn}(z)B^{+}, \text{ иначе.} \end{cases}$$

Описание наиболее часто используемых машинно реализуемых функций представлено в Приложении 1 в виде программного кода на языке Free Pascal. Именно в таком виде используются функции для формирования алфавита в разработанных методах символьной регрессии и в разработанных программных комплексах для их реализации.

Теорема 3.2. *Любую машинно реализуемую функцию можно представить в виде ряда Тейлора с конечным числом членов.*

Доказательство.

Предположим, что $f(z)$ – машинно реализуемая функция, тогда для точки $z = a$ ряд Тейлора имеет следующий вид:

$$\sum_{\{k=0\}}^{L} \frac{f^{(k)}(a)}{k!}(z-a)^k = f(a) + f'(a)(z-a) + \frac{f''(a)}{2!}(z-a)^2 + \cdots$$

$$\ldots + \frac{f^{(L)}(a)}{L!}(z-a)^L.$$

Значение производной ограничено $\left|f^{(k)}(a)\right| \leq B^{+}$, а значение знаменателя возрастает $k!$. Для некоторого члена ряда Тейлора будет выполняться следующее неравенство

$$\frac{B^{+}}{k!} < \delta^{-}$$

Согласно свойству (3.86), все последующие члены ряда будут равны нулю. ■.

Для поиска математических выражений в пространстве машинно реализуемых функций можно использовать известные численные методы, в том числе и эволюционные алгоритмы машинного обучения.

3.7. Вариационные методы символьной регрессии

Все методы символьной регрессии различаются в зависимости от формы кодирования математического выражения, так же в них различаются и основные операции генетического алгоритма – скрещивание и мутация. Таким образом, для описания методов символьной регрессии или создания новых методов символьной регрессии необходимо определить форму кодирования возможного решения и переопределить операции скрещивания и мутации генетического алгоритма так, чтобы в результате выполнения этих операций получались правильные коды новых возможных решений.

Некоторые успешные решения прикладных задач управления до сих пор демонстрировались главным образом методом генетического программирования [196, 197]. Сегодня разработаны и другие методы символьной регрессии, которые могут справиться с решением задач машинного обучения управления.

Методы символьной регрессии позволяют автоматизировать процесс синтеза систем управления, но очень мало используются в этом направлении ввиду ряда трудностей, таких как нечисловое пространство поиска и отсутствие метрики нем, сложность программного кода и отсутствие общедоступных программных пакетов и т. д. Разработкой этих подходов специально для решения задачи синтеза систем управления занимаются пока только три научные группы в мире: одну возглавляет Р. Бабушка (Чехия) [185, 198], вторую – Б. Ноак (Китай) [142, 199, 200], а другая – это наша научная группа, возглавляемая профессором А.И. Дивеевым (Россия) [141, 144]. Европейская и китайская исследовательские группы ищут функции управления как линейные комбинации основных функций, причем в качестве основных функций используются в основном гладкие функции. Мы же делаем синтез в виде вложения функций, что позволяет получать более сложные математические выражения, а также использовать более широкий набор базовых функций, в том числе разрывные функции, что позволяет представить if-оператор условия, который часто требуется при создании сложных интеллектуальных функций управления.

Получать успешные решения задач синтеза систем управления позволяет применение принципа малых вариаций базисного решения, представленного в разделе 3.4. Применение принципа малых вариаций к различным методам символьной регрессии не является прихотью автора. Опыт изучения применения методов символьной регрессии для решения сложных задач управления показал, что только вариационные методы символьной регрессии позволяют получить приемлемые решения за разумное время. Детальное изучение этого вопроса, вероятно, еще впереди, и оно связано с выяснением глобальных причин успеха эволюционных вычислений.

В данном разделе диссертационного исследования представим несколько методов символьной регрессии, модифицированных с помощью принципа малых вариаций базисного решения в рамках диссертационного исследования и успешно примененных для решения задач синтеза управления.

- Метод сетевого оператора
- Вариационное генетическое программирование
- Вариационное аналитическое программирование
- Вариационное Декартово генетическое программирование
- Вариационное полное бинарное генетическое программирование
- Метод многослойного сетевого оператора

Применение принципа малых вариаций базисного решения к каждому конкретному методу требует определения типов малых вариаций и способов их кодирования. Рассмотрим предложенные подходы.

3.7.1. Метод сетевого оператора

Первым методом, в котором был использован принцип малых вариаций базисного решения, был метод сетевого оператора [195], разработанный Дивеевым А.И. Этот метод разрабатывался непосредственно с целью решения задач синтеза управления, поэтому изначально учитывал описанную специфику данных задач. Еще в 2008 году в рамках кандидатский диссертации автору

удалось реализовать с помощью этого метода численное решение задачи синтеза из области начальных условий [201, 202]

Метод сетевого оператора кодирует математическое выражение в виде ориентированного графа. Метод использует только функции с одним и двумя аргументами. В ориентированном графе сетевого оператора аргументы математического выражения связаны с исходными узлами, функции с одним аргументом связаны с дугами графа, функции с двумя аргументами связаны с другими узлами графа. Любые узлы графа можно определить как выходы и результатами вычислений в них будут значения компонент выходного вектора. Узлы-стоки в графе являются выходными данными математического выражения.

Для кодирования математического выражения методом сетевого оператора используются три упорядоченных множества

- множество аргументов математического выражения, или множество параметров и переменных математического выражения

$$\mathrm{F}_0 = \{f_{0,1} = q_1, \dots, f_{0,p} = q_p, f_{0,p+1} = x_1, \dots, f_{0,n+p} = x_n\}. \quad (3.90)$$

- множество функций с одним аргументом

$$\mathrm{F}_1 = \{f_{1,1} = z, f_{1,2}(z), \dots, f_{1,w}(z)\}, \quad (3.91)$$

- множество функций с двумя аргументами

$$\mathrm{F}_2 = \{f_{2,1}(z_1, z_2), \dots, f_{2,v}(z_1, z_2)\}. \quad (3.92)$$

Набор функций с одним аргументом F_1 обязательно должен содержать тождественную функцию:

$$f_{1,1} = z.$$

Функции с двумя аргументами должны быть ассоциативными, коммутативными и иметь единичный элемент.

Для построения вычислительного графа сетевого оператора необходимо представить математическое выражение в виде композиции функций. Первая функция в композиции должна быть функцией с двумя аргументами. Далее в

записи композиции функции с одним аргументом чередуются с функциями с двумя аргументами.

Также при построении графа сетевого оператора также необходимо соблюдать следующие правила:

- аргументами функций с одним аргументом являются только функции с двумя аргументами или элементы из множества параметров и переменных,
- аргументами функции с двумя аргументами могут быть только функции с одним аргументом или единичный элемент этой функции,
- функции с двумя аргументами не должны включать функции с тем же аргументом в качестве аргументов.

Для удовлетворения этих требований к построению графа сетевого оператора вводятся дополнительные функции с двумя аргументами и с единичным элементом этой функции в качестве второго аргумента. Единичный элемент на графе не указан. Если узел графа содержит одну дугу, то второй аргумент, связанный с этим узлом, является единичным элементом, поэтому функция с двумя аргументами, связанная с узлом только с одной дугой, не изменяет входное значение.

Рассмотрим пример.

Пусть задано математическое выражение

$$y = (x_1^2 - x_2^2)\cos(q_1 x_1 + q_2) + x_1 x_2 e^{-q_3 x_1}. \qquad (3.93)$$

Для кодирования этого математического выражения необходимы следующие множества аргументов и функций:

$\mathrm{F}_0 = \{x_1, x_2, q_1, q_2, q_3\}$,

$\mathrm{F}_1 = \{f_{1,1}(z) = z, f_{1,2}(z) = -z, f_{1,3}(z) = \cos(z), f_{1,4}(z) = e^z, f_{1,5}(z) = z^2\}$,

$\mathrm{F}_2 = \{f_{2,1}(z_1, z_2) = z_1 + z_2, f_{2,2}(z_1, z_2) = z_1 z_2\}$.

Граф сетевого оператора математического выражения (3.93) показан на рис. 3.3.

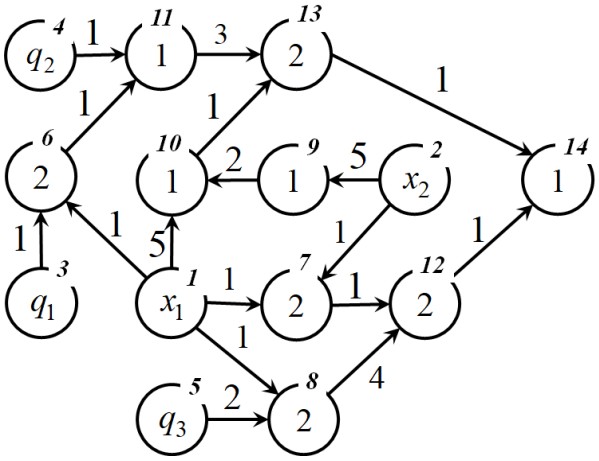

Рис.3.3. Граф сетевого оператора математического выражения (3.93)

На рис. 3.3 в узлах указаны номера функций с двумя аргументами, рядом с дугами указаны номера функций с одним аргументом, в узлах-источниках указаны аргументы математического выражения, а сама нумерация узлов записана в верхних частях узлов. Узлы нумеруются по правилам топологической сортировки, номер узла, из которого выходит дуга, меньше номера узла, в который входит дуга.

В памяти компьютера граф сетевого оператора представляется в виде целочисленной квадратной верхнетреугольной матрицы сетевого оператора, которая строится из матрицы смежности графа. В графе сетевого оператора вместо единиц, которые были в матрице смежности, указаны номера функций с одним аргументом, на диагонали указаны номера функций с двумя аргументами.

Матрица сетевого оператора для графа математического выражения (3.93) имеет следующий вид

$$\Psi = \begin{bmatrix}
0&0&0&0&0&1&1&1&0&5&0&0&0&0\\
0&0&0&0&0&0&1&0&5&0&0&0&0&0\\
0&0&0&0&0&1&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&1&0&0&0\\
0&0&0&0&0&0&0&2&0&0&0&0&0&0\\
0&0&0&0&0&2&0&0&0&0&1&0&0&0\\
0&0&0&0&0&0&2&0&0&0&0&1&0&0\\
0&0&0&0&0&0&0&2&0&0&0&4&0&0\\
0&0&0&0&0&0&0&0&1&2&0&0&0&0\\
0&0&0&0&0&0&0&0&0&1&0&0&1&0\\
0&0&0&0&0&0&0&0&0&0&1&0&3&0\\
0&0&0&0&0&0&0&0&0&0&0&2&0&1\\
0&0&0&0&0&0&0&0&0&0&0&0&2&1\\
0&0&0&0&0&0&0&0&0&0&0&0&0&1
\end{bmatrix}. \quad (3.94)$$

Сложность построения графа сетевого оператора по математическому выражению компенсируется тем, что такое построение не требуется выполнять при поиске математического выражения. Матрица сетевого оператора, согласно методу, задается один раз только для базисного решения. Это может быть определенная заданная матрица, описывающая конкретное математическое выражение, например, П-регулятор, а может быть любая целочисленная квадратная верхнетреугольная матрица, в которой нулевые столбцы соответствуют исходным узлам, а в остальных столбцах и во всех строках недиагональные ненулевые элементы, значения которых соответствуют номерам функций с одним аргументом, а по диагонали в ненулевых столбцах стоят номера функций с двумя аргументами, то есть любая матрица сетевого оператора, описывающая некоторый вычислительный граф.

Наиболее часто задействованной процедурой является расчет результата математического выражения по матрице сетевого оператора. Данная процедура реализуется как внутри оптимизационного алгоритма поиска для вычисления значения управления, так и как самостоятельный бортовой программный модуль, расположенный на объекте управления и выдающий по матрице сетевого оператора, полученной в результате решения задачи синтеза, значения управления для объекта в каждый момент времени. Поэтому такая процедура реализована программно как внутри общего метода сетевого оператора, так и в качестве самостоятельного модуля [203].

Рассмотрим пример вычисления математического выражения с использованием графа сетевого оператора.

Для хранения результатов промежуточных вычислений вводится вектор узлов, размерность которого равна количеству строк матрицы сетевого оператора.

- Инициализируем вектор узлов и помещаем значения аргументов и единичных элементов функций с двумя аргументами в его соответствующие компоненты.

$$\mathbf{z}^{(0)} = [x_1\ x_2\ q_1\ q_2\ q_3\ 1\ 1\ 1\ 0\ 0\ 0\ 1\ 1\ 0]^T. \qquad (3.95)$$

- Сканируем матрицу (3.96) построчно и находим ненулевые недиагональные элементы $\psi_{i,j} \neq 0, i = 1, \dots, 13, j = i + 1, \dots, 14$.

$\psi_{1,6} = 1, \quad z_6^{(1)} = f_{2,2}(z_6^{(0)}, f_{1,1}(x_1)) = 1 \cdot x_1 = x_1,$

$\psi_{1,7} = 1, \quad z_7^{(1)} = f_{2,2}\left(z_7^{(0)}, f_{1,1}(x_1)\right) = 1 \cdot x_1 = x_1,$

$\psi_{1,8} = 1, \quad z_8^{(1)} = f_{2,2}\left(z_8^{(0)}, f_{1,1}(x_1)\right) = 1 \cdot x_1 = x_1,$

$\psi_{1,10} = 5, \quad z_{10}^{(1)} = f_{2,2}\left(z_{10}^{(0)}, f_{1,5}(x_1)\right) = 1 \cdot x_1^2 = x_1^2,$

$$z_i^{(1)} = z_i^{(0)}, \quad i = 1, \dots, 5, 9, 11, \dots, 14,$$

$\psi_{2,7} = 1, \quad z_7^{(2)} = f_{2,2}\left(z_7^{(1)}, f_{1,1}(x_2)\right) = x_1 \cdot x_2 = x_1 x_2,$

$\psi_{2,9} = 5, \quad z_9^{(2)} = f_{2,2}\left(z_9^{(1)}, f_{1,5}(x_2)\right) = 1 \cdot x_2^2 = x_2^2,$

$$z_i^{(2)} = z_i^{(1)}, \quad i = 1, \dots, 6, 8, 10, \dots, 14,$$

$\psi_{3,6} = 1, \quad z_6^{(3)} = f_{2,2}\left(z_6^{(2)}, f_{1,1}(q_1)\right) = x_1 \cdot q_1 = x_{1q_1},$

$$z_i^{(3)} = z_i^{(2)}, \quad i = 1, \dots, 5, 7, \dots, 14,$$

$\psi_{4,11} = 1, \quad z_{11}^{(4)} = f_{2,1}\left(z_{11}^{(3)}, f_{1,1}(q_2)\right) = 0 + q_2 = q_2,$

$$z_i^{(4)} = z_i^{(3)}, \quad i = 1, \dots, 10, 12, 13, 14,$$

$\psi_{5,8} = 2, \quad z_8^{(5)} = f_{2,2}\left(z_8^{(4)}, f_{1,2}(q_3)\right) = 1 \cdot (-q_3) = -q_3,$

$$z_i^{(5)} = z_i^{(4)}, \quad i = 1, \dots, 7, 9, \dots, 14,$$

$$\psi_{6,11}=2,\quad z_{11}^{(6)}=f_{2,1}\left(z_{11}^{(5)},f_{1,1}\left(z_{6}^{(5)}\right)\right)=q_2+x_1q_1,$$

$$z_i^{(6)}=z_i^{(5)},\quad i=1,\dots,10,12,13,14,$$

$$\psi_{7,12}=1,\quad z_{12}^{(7)}=f_{2,2}\left(z_{12}^{(6)},f_{1,1}\left(z_{7}^{(6)}\right)\right)=x_1x_2,$$

$$z_i^{(7)}=z_i^{(6)},\quad i=1,\dots,11,13,14,$$

$$\psi_{8,12}=4,\quad z_{12}^{(8)}=f_{2,2}\left(z_{12}^{(7)},f_{1,4}\left(z_{8}^{(7)}\right)\right)=e^{-x_{1q_3}},$$

$$z_i^{(8)}=z_i^{(7)},\quad i=1,\dots,11,13,14,$$

$$\psi_{9,10}=2,\quad z_{10}^{(9)}=f_{2,1}\left(z_{10}^{(8)},f_{1,2}\left(z_{9}^{(8)}\right)\right)=x_1^2-x_2^2,$$

$$z_i^{(9)}=z_i^{(8)},\quad i=1,\dots,9,11,12,13,14,$$

$$\psi_{10,13}=1,z_{13}^{(10)}=f_{2,2}\left(z_{13}^{(9)},f_{1,1}\left(z_{10}^{(9)}\right)\right)=x_1^2-x_2^2,$$

$$z_i^{(10)}=z_i^{(9)},\quad i=1,\dots,12,14,$$

$$\psi_{11,13}=3,\quad z_{13}^{(11)}=f_{2,2}\left(z_{13}^{(10)},f_{1,2}\left(z_{11}^{(10)}\right)\right)=(x_1^2-x_2^2)\cos(x_1q_1+q_2),$$

$$z_i^{(11)}=z_i^{(10)},\quad i=1,\dots,12,14,$$

$$\psi_{12,14}=1,\quad z_{14}^{(12)}=f_{2,1}\left(z_{14}^{(11)},f_{1,1}\left(z_{12}^{(11)}\right)\right)=x_1x_2e^{-x_{1q_3}},$$

$$z_i^{(12)}=z_i^{(11)},\quad i=1,\dots,13,$$

$$\psi_{13,14}=1,z_{14}^{(13)}=f_{2,1}\left(z_{14}^{(12)},f_{1,1}\left(z_{13}^{(12)}\right)\right)=$$

$$=(x_1^2-x_2^2)\cos(x_1q_1+q_2)+x_1x_2e^{-x_1q_3}$$

$$z_i^{(13)}=z_i^{(12)},\quad i=1,\dots,13.$$

Как только все строки матрицы сетевого оператора просканированы, последний элемент вектора узлов содержит результат вычисления математического выражения, закодированного сетевым оператором.

Метод сетевого оператора является первым методом символьной регрессии, который построен на принципе малых вариаций базисного решения.

Согласно принципу, генетические операции выполняются не непосредственно над матрицей сетевого оператора, а над множеством вариаций.

Только одно базисное решение закодировано в виде матрицы сетевого оператора. Другие возможные решения представлены как небольшие вариации этого базисного решения. Для кодирования всех возможных решений используется множество векторов малых вариаций.

В методе сетевого оператора каждый вектор малой вариации включает четыре целочисленных компоненты:

$$\mathbf{w} = [w_1 \;\; w_2 \;\; w_3 \;\; w_4]^T, \tag{3.96}$$

где w_1 - тип вариации, w_2 - номер строки матрицы сетевого оператора, w_3 - номер столбца матрицы сетевого оператора, w_4 - новое значение элемента в матрице сетевого оператора.

Для матрицы сетевого оператора используются следующие типы малых вариаций:

$w_1 = 0$ - замена ненулевого недиагонального элемента,

$w_1 = 1$ - замена ненулевого диагонального элемента,

$w_1 = 2$ - замена нулевого недиагонального элемента ненулевым элементом,

$w_1 = 3$ - замена ненулевого недиагонального элемента нулевым элементом.

Вариации выполняются при сохранении условия корректности матрицы сетевого оператора:

$$\forall i \; \exists \psi_{i,j} \neq 0, i \neq j \;\; \text{и} \; \forall j \;\; \notin \{ s_1, \dots, s_{N+P} \}, \;\; \exists \psi_{k,j} \neq 0, \;\; k \;\; \neq j. \tag{3.97}$$

Условие (3.97) корректности матрицы сетевого оператора означает отсутствие таких строк и столбцов, не соответствующих исходным узлам, в которых все недиагональные элементы равны нулю.

Рассмотрим пример.

Пусть задан следующий вектор вариаций:

$$\mathbf{w} = [2 \;\; 8 \;\; 10 \;\; 3]^T. \tag{3.98}$$

После этой малой вариации матрица сетевого оператора (3.94) имеет следующий вид

$$\mathbf{w} \circ \Psi = \begin{bmatrix} 0\,0\,0\,0\,0\,1\,1\,1\,0\,5\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0\,0\,1\,0\,5\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,1\,0\,0\,0\,0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,0 \\ 0\,0\,0\,0\,0\,0\,0\,2\,0\,0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,2\,0\,0\,0\,0\,1\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0\,2\,0\,0\,0\,0\,1\,0\,0 \\ 0\,0\,0\,0\,0\,0\,0\,2\,0\,3\,0\,4\,0\,0 \\ 0\,0\,0\,0\,0\,0\,0\,0\,1\,2\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,0\,1\,0 \\ 0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,3\,0 \\ 0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,2\,0\,1 \\ 0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,2\,1 \\ 0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1 \end{bmatrix}. \tag{3.99}$$

Эта новая матрица сетевого оператора кодирует следующее математическое выражение

$$\tilde{y} = (x_1^2 - x_2^2 + \cos(-q_3 x_1))\cos(x_1 q_1 + q_2) + x_1 x_2 e^{-x_1 q_3}. \tag{3.100}$$

Теперь рассмотрим генетические операции в методе сетевого оператора.

Каждое возможное решение кодируется набором векторов малых вариаций:

$$\mathrm{W}_k = (\mathbf{w}^{k,1}, \dots, \mathbf{w}^{k,d}), \tag{3.101}$$

где d - количество малых вариаций или глубина вариаций.

Кроссовер выполняется для наборов векторов малых вариаций. Для этого выбираются два возможных решения

$$\mathrm{W}_\alpha = (\mathbf{w}^{\alpha,1}, \dots, \mathbf{w}^{\alpha,d}), \tag{3.102}$$

$$\mathrm{W}_\beta = (\mathbf{w}^{\beta,1}, \dots, \mathbf{w}^{\beta,d}). \tag{3.103}$$

Случайным образом определяем точку скрещивания $c \in \{1, \dots, d\}$.

Два новых возможных решения получаются путем обмена векторами вариаций, стоящих после точки скрещивания

$$\mathrm{W}_\gamma = \left(\mathbf{w}^{\alpha,1}, \dots, \mathbf{w}^{\alpha,c}, \mathbf{w}^{\beta,c+1}, \dots, \mathbf{w}^{\beta,d}\right), \tag{3.104}$$

$$\mathrm{W}_\delta = \left(\mathbf{w}^{\beta,1}, \dots, \mathbf{w}^{\beta,c}, \mathbf{w}^{\alpha,c+1}, \dots, \mathbf{w}^{\alpha,d}\right). \tag{3.105}$$

Мутация выполняется для новых возможных решений (3.104), (3.105) путем случайной генерации новых векторов вариации в случайно выбранной позиции.

Данный подход программно реализован и успешно применен во многих работах автора для синтеза систем управления [204 – 206].

3.7.2. Метод вариационного генетического программирования

Метод генетического программирования является первым и на сегодняшний день самым известным методом символьной регрессии. Он был создан еще в 90-х годах XX века Дж. Козой [207].

Математическое выражение в генетическом программировании представляет собой строку символов без круглых скобок. Каждый символ соответствует некоторой элементарной функции или аргументу математического выражения. В качестве символов при поиске математического выражения используются двухкомпонентные числовые векторы. Первая компонента вектора указывает количество аргументов кодируемой элементарной функции, а вторая компонента указывает номер функции в заданном алфавите функций. Эти два числа необходимы для вычисления математического выражения с использованием кода генетического программирования и для нахождения подвыражения, используемого при скрещивании. Функции без аргументов являются переменными или параметрами. Порядок символов в строке определяет соответствие между функциями и их аргументами. В генетическом программировании чаще используется префиксная запись символов. В этой записи символ функции появляется в строке до или слева от символа аргумента.

Рассмотрим процедуру кодирования математического выражения согласно методу генетического программирования.

Введем множество упорядоченных наборов функций с определенным числом аргументов:

$$\mathrm{F} = \{\mathrm{F}_0, \mathrm{F}_1, \dots, \mathrm{F}_n\}, \tag{3.106}$$

где

$$\mathrm{F}_i = \{f_{i,1}(z_1, \dots, z_i), \dots, f_{i,n_i}(z_1, \dots, z_i)\},$$

$f_{i,j}(z_1, \dots, z_i)$ - номер функции j с количеством аргументов i, $j = 1, \dots, n_i$, $i = 1, \dots, n$. Множество F_0 является множеством аргументов математического выражения, которые рассматриваются как элементарные функции без аргументов.

Код функции представляет собой целочисленный вектор из двух компонент

$$\mathbf{s} = [s_1\ s_2]^T, \tag{3.107}$$

где s_1 - количество аргументов, s_2 - номер функции в множестве F_{s_1},

$$f_{i,j}(z_1, \dots, z_i) \Leftrightarrow [i\ \ j]^T.$$

Математическое выражение представляет собой упорядоченный набор кодов функций:

$$\mathrm{S} = (\mathbf{s}^1, \dots, \mathbf{s}^K), \tag{3.108}$$

где $\mathbf{s}^i = \left[s_1^i\ \ s_2^i\right]^T$, $i = 1, \dots, K$.

Рассмотрим пример.

Пусть задано следующее математическое выражение:

$$y_1 = e^{-ax_1}\cos(bx_2 + c). \tag{3.109}$$

Определим алфавит элементарных функций для этого примера:

$$\mathrm{F}_0 = \{f_{0,1} = x_1, f_{0,2} = x_2, f_{0,3} = a, f_{0,4} = b, f_{0,5} = c\}, \tag{3.110}$$
$$\mathrm{F}_1 = \{f_{1,1}(z) = -z, f_{1,2}(z) = e^z, f_{1,3}(z) = \cos(z)\},$$
$$\mathrm{F}_2 = \{f_{2,1}(z_1, z_2) = z_1 + z_2, f_{2,2}(z_1, z_2) = z_1 z_2\}.$$

Перепишем множеств0 функций (3.110) в кодах

$$\mathrm{F}_0 = \left(\begin{bmatrix}0\\1\end{bmatrix}, \begin{bmatrix}0\\2\end{bmatrix}, \begin{bmatrix}0\\3\end{bmatrix}, \begin{bmatrix}0\\4\end{bmatrix}, \begin{bmatrix}0\\5\end{bmatrix}\right), \tag{3.111}$$
$$\mathrm{F}_1 = \left(\begin{bmatrix}1\\1\end{bmatrix}, \begin{bmatrix}1\\2\end{bmatrix}, \begin{bmatrix}1\\3\end{bmatrix}\right),$$
$$\mathrm{F}_2 = \left(\begin{bmatrix}2\\1\end{bmatrix}, \begin{bmatrix}2\\2\end{bmatrix}\right).$$

Тогда математическое выражение (3.109) будет кодироваться следующим образом:

$$y_1 = e^{-ax_1}\cos(bx_2 + c) = f_{2,2}\left(e^{-ax_1}, \cos(bx_2 + c)\right) =$$

$$f_{2,2}\left(e^{-ax_1}, \cos\left(f_{2,1}(bx_2, c)\right)\right) = f_{2,2}\left(f_{1,2}(-ax_1), f_{1,3}\left(f_{2,1}(bx_2, c)\right)\right) =$$

$$f_{2,2}(f_{1,2}(f_{2,2}(-a, x_1)), f_{1,3}(f_{2,1}(f_{2,2}(b, x_2), c))) =$$

$$f_{2,2}\left(f_{1,2}\left(f_{2,2}(f_{1,1}(f_{0,3}), f_{0,1})\right), f_{1,3}\left(f_{2,1}(f_{2,2}(f_{0,4}, f_{0,2}), f_{0,5})\right)\right).$$

В кодах (3.111) оно запишется следующим образом:

$$S = \left(\begin{bmatrix}2\\2\end{bmatrix}, \begin{bmatrix}1\\2\end{bmatrix}, \begin{bmatrix}2\\2\end{bmatrix}, \begin{bmatrix}1\\1\end{bmatrix}, \begin{bmatrix}0\\3\end{bmatrix}, \begin{bmatrix}0\\1\end{bmatrix}, \begin{bmatrix}1\\3\end{bmatrix}, \begin{bmatrix}2\\1\end{bmatrix}, \begin{bmatrix}2\\2\end{bmatrix}, \begin{bmatrix}0\\2\end{bmatrix}, \begin{bmatrix}0\\4\end{bmatrix}, \begin{bmatrix}0\\5\end{bmatrix}\right), \quad (3.112)$$

На рис. 3.4 представлено дерево вычислений для заданного математического выражения.

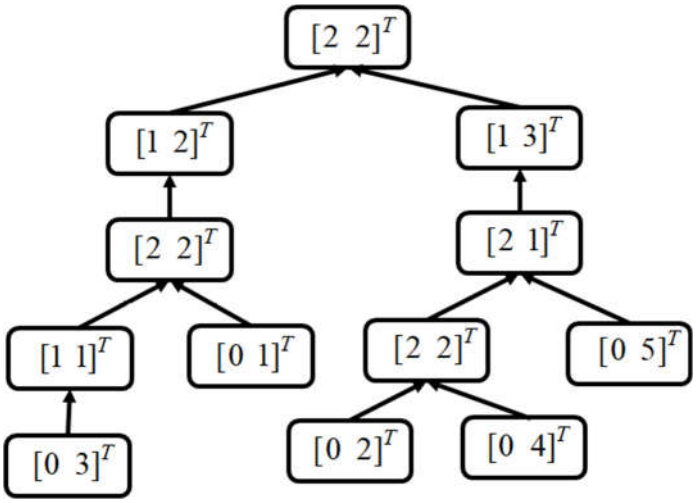

Рис.3.4. Дерево вычислений математического выражения (3.109) в методе генетического программирования

Основной вычислительный недостаток метода генетического программирования состоит в том, что разные математические выражения имеют разную длину кода и, соответственно, после операции скрещивания длина кода постоянно меняется.

В этой связи принцип малых вариаций, примененный к методу генетического программирования, дополнительно позволяет преодолеть и указанную вычислительную проблему.

Определим малые вариации кода генетического программирования:

$w_1 = 1$ - изменение второй компоненты вектора кода функции, которая указывает на индекс элемента из множества, заданного первой компонентой;

$w_1 = 2$ - удаление функции с одним аргументом;

$w_1 = 3$ - вставка функции с одним аргументом;

$w_1 = 4$ - увеличение значения первой компоненты вектора кода функции, при этом вектор кода аргумента вставляется после кода функции;

$w_1 = 5$ - уменьшение значения первой компоненты вектора кода функции на единицу, при этом удаляется первый код аргумента, встречающийся после кода функции.

Таким образом, малая вариация для кода в генетическом программировании может быть описана следующим вектором вариации из трех компонент:

$$\mathbf{w} = [w_1\ w_2\ w_3]^T, \tag{3.113}$$

где w_1 - тип вариации, $w_1 \in \{1, \dots, 5\}$, w_2 - индекс изменяемого элемента, w_3 - значение нового элемента.

Рассмотрим пример. Возьмем выражение (3.111), закодированное в (3.112), и алфавит элементарных функций (3.110), расширенный новыми функциями для дальнейшей демонстрации вариаций.

$$\mathrm{F}_0 = \{f_{0,1} = x_1, f_{0,2} = x_2, f_{0,3} = a, f_{0,4} = b, f_{0,5} = c\},$$

$$\mathrm{F}_1 = \{f_{1,1}(z) = -z, \quad f_{1,2}(z) = e^z, \ f_{1,3}(z) = \cos(z), \ f_{1,4}(z) = \sin(z), \ f_{1,5}(z) = z^2\},$$

$$\mathrm{F}_2 = \{f_{2,1}(z_1, z_2) = z_1 + z_2, f_{2,2}(z_1, z_2) = z_1 z_2\}.$$

Зададим базисное решение S_0 в виде (3.112):

$$\mathrm{S}_0 = \left(\begin{bmatrix}2\\2\end{bmatrix}, \begin{bmatrix}1\\2\end{bmatrix}, \begin{bmatrix}2\\2\end{bmatrix}, \begin{bmatrix}1\\1\end{bmatrix}, \begin{bmatrix}0\\3\end{bmatrix}, \begin{bmatrix}0\\1\end{bmatrix}, \begin{bmatrix}1\\3\end{bmatrix}, \begin{bmatrix}2\\1\end{bmatrix}, \begin{bmatrix}2\\2\end{bmatrix}, \begin{bmatrix}0\\2\end{bmatrix}, \begin{bmatrix}0\\4\end{bmatrix}, \begin{bmatrix}0\\5\end{bmatrix}\right), \tag{3.114}$$

Пусть возможные решения заданы в виде упорядоченных множеств векторов малых вариаций:

$$\mathrm{W}_1 = (\mathbf{w}^{1,1}, \dots, \mathbf{w}^{1,4}) = \left(\begin{bmatrix}1\\3\\1\end{bmatrix}, \begin{bmatrix}3\\10\\3\end{bmatrix}, \begin{bmatrix}4\\2\\1\end{bmatrix}, \begin{bmatrix}5\\1\\2\end{bmatrix}\right) \tag{3.115}$$

$$\mathrm{W}_2 = (\mathbf{w}^{2,1}, \dots, \mathbf{w}^{2,4}) = \left(\begin{bmatrix}1\\1\\1\end{bmatrix}, \begin{bmatrix}3\\4\\4\end{bmatrix}, \begin{bmatrix}3\\2\\5\end{bmatrix}, \begin{bmatrix}2\\9\\3\end{bmatrix}\right) \quad (3.116)$$

Для получения вариации кодов берется первый вектор вариации $\mathbf{w}^{1,1} = [1\ \ 3\ \ 1]^T$. Первая компонента $w_1^{1,1} = 1$, это означает, что необходимо заменить вторую компоненту в векторе кода $w_2^{1,1} = 3$ на значение $w_2^{1,1} = 1$. Результат: $\mathbf{s}^{1,3} = [2\ \ 1]^T$. Далее выполняются все остальные вариации.

В результате коды для этих возможных решений будут иметь следующий вид:

$$\mathrm{S}_1 = \mathbf{w}^{1,4} \circ \mathbf{w}^{1,3} \circ \mathbf{w}^{1,2} \circ \mathbf{w}^{1,1} \circ \mathrm{S}_0 =$$

$$\left(\begin{bmatrix}1\\2\end{bmatrix}, \begin{bmatrix}2\\2\end{bmatrix}, \begin{bmatrix}2\\1\end{bmatrix}, \begin{bmatrix}1\\1\end{bmatrix}, \begin{bmatrix}0\\3\end{bmatrix}, \begin{bmatrix}0\\1\end{bmatrix}, \begin{bmatrix}1\\3\end{bmatrix}, \begin{bmatrix}2\\1\end{bmatrix}, \begin{bmatrix}2\\2\end{bmatrix}, \begin{bmatrix}1\\3\end{bmatrix}, \begin{bmatrix}0\\2\end{bmatrix}, \begin{bmatrix}0\\4\end{bmatrix}, \begin{bmatrix}0\\5\end{bmatrix}\right), \quad (3.117)$$

$$\mathrm{S}_2 = \mathbf{w}^{2,4} \circ \mathbf{w}^{2,3} \circ \mathbf{w}^{2,2} \circ \mathbf{w}^{2,1} \circ \mathrm{S}_0 =$$

$$\left(\begin{bmatrix}2\\1\end{bmatrix}, \begin{bmatrix}1\\5\end{bmatrix}, \begin{bmatrix}1\\2\end{bmatrix}, \begin{bmatrix}2\\2\end{bmatrix}, \begin{bmatrix}1\\4\end{bmatrix}, \begin{bmatrix}1\\1\end{bmatrix}, \begin{bmatrix}0\\3\end{bmatrix}, \begin{bmatrix}0\\1\end{bmatrix}, \begin{bmatrix}2\\1\end{bmatrix}, \begin{bmatrix}2\\2\end{bmatrix}, \begin{bmatrix}0\\2\end{bmatrix}, \begin{bmatrix}0\\4\end{bmatrix}, \begin{bmatrix}0\\5\end{bmatrix}\right). \quad (3.118)$$

Заметим, что, если при исполнении вариации возникает какое-то противоречие, то малая вариация не исполняется. Так при исполнении вариации $\mathbf{w}^{2,4} = [2\ \ 9\ \ 3]^T$ первая компонента $w_1^{2,4} = 2$ означает, что необходимо удалить функцию с одним аргументом в позиции кода $w_2^{2,4} = 9$. Но такой компоненты в данной позиции в коде нет, поэтому вариация просто не исполняется.

Коды возможных решений (3.117) и (3.118) соответствуют математическим выражениям

$$y_1 = e^{\left((-a+x_1)\cos(b\cos(x_2)+c)\right)},$$

$$y_2 = e^{2(x_1\sin(-a))} + bx_2 + c.$$

Теперь покажем схему выполнения операции скрещивания. Она выполняется по классической схеме, но для возможных решений, заданных в виде наборов векторов вариаций (3.115), (3.116).

Выбираем точку скрещивания $c \in \{1, \dots, 4\}$. Например, $c = 3$.

Новые возможные решения будут следующими:

$$\mathrm{W}_3 = (\mathbf{w}^{1,1}, \mathbf{w}^{1,2}, \mathbf{w}^{2,3}, \mathbf{w}^{2,4}), \quad (3.119)$$

$$\mathrm{W}_4 = (\mathbf{w}^{2,1}, \mathbf{w}^{2,2}, \mathbf{w}^{1,3}, \mathbf{w}^{1,4}). \quad (3.120)$$

Применяя полученные вектора малых вариаций (3.119), (3.120) к базисному решению, получаем следующие коды:

$$S_3 = \mathbf{w}^{3,4} \circ \mathbf{w}^{3,3} \circ \mathbf{w}^{3,2} \circ \mathbf{w}^{3,1} \circ S_0 =$$

$$\left(\begin{bmatrix}2\\2\end{bmatrix},\begin{bmatrix}1\\5\end{bmatrix},\begin{bmatrix}1\\2\end{bmatrix},\begin{bmatrix}2\\1\end{bmatrix},\begin{bmatrix}1\\1\end{bmatrix},\begin{bmatrix}0\\3\end{bmatrix},\begin{bmatrix}0\\1\end{bmatrix},\begin{bmatrix}1\\3\end{bmatrix},\begin{bmatrix}2\\1\end{bmatrix},\begin{bmatrix}2\\2\end{bmatrix},\begin{bmatrix}1\\3\end{bmatrix},\begin{bmatrix}0\\2\end{bmatrix},\begin{bmatrix}0\\4\end{bmatrix},\begin{bmatrix}0\\5\end{bmatrix}\right), \quad (3.121)$$

$$S_4 = \mathbf{w}^{4,4} \circ \mathbf{w}^{4,3} \circ \mathbf{w}^{4,2} \circ \mathbf{w}^{4,1} \circ S_0 =$$

$$\left(\begin{bmatrix}1\\5\end{bmatrix},\begin{bmatrix}2\\2\end{bmatrix},\begin{bmatrix}2\\2\end{bmatrix},\begin{bmatrix}1\\4\end{bmatrix},\begin{bmatrix}1\\1\end{bmatrix},\begin{bmatrix}0\\3\end{bmatrix},\begin{bmatrix}0\\1\end{bmatrix},\begin{bmatrix}1\\3\end{bmatrix},\begin{bmatrix}2\\1\end{bmatrix},\begin{bmatrix}2\\2\end{bmatrix},\begin{bmatrix}0\\2\end{bmatrix},\begin{bmatrix}0\\4\end{bmatrix},\begin{bmatrix}0\\5\end{bmatrix}\right). \quad (3.122)$$

Новые коды (3.121) и (3.122) соответствуют следующим математическим выражениям

$$y_3 = e^{-a+x_1} + \cos(b\cos(x_2) + c),$$

$$y_4 = \left(x_1 \sin(-a)\cos(bx_2 + c)\right)^2.$$

Метод вариационного генетического программирования также программно реализован и успешно применен во работах автора для синтеза систем управления [208, 209].

Многие исследователи отмечают некоторые недостатки традиционного генетического программирования, такие как необходимость рекурсивных вычислений при поиске подвыражения для скрещивания и разная длина кодов для разных выражений, вызывающая дополнительные вычислительные трудности. Этот факт послужил толчком к развитию других методов кодирования, направленных на преодоление этих трудностей, и соответственно новых методов символьной регрессии.

3.7.3. Метод вариационного аналитического программирования

Еще одним методом символьной регрессии, для которого был применен принцип малых вариаций, является метод аналитического программирования [210, 211]. Аналитическое программирование является разновидностью метода генетического программирования. Отличие кодов математических выражений состоит в том, что в аналитическом программировании количество аргументов функции не указывается напрямую, а определяется количеством функций. Отсюда следует, что можно применять малые вариации и вектор кодирования

этих вариаций аналогично методу вариационного генетического программирования, но только базисное решение будет иметь другую кодировку.

Элементы математического выражения в аналитическом программировании кодируются целыми числами. Все элементы математического выражения вместе с аргументами собираются в одно упорядоченное множество. Каждое число в упорядоченном множестве является кодом элемента.

В качестве алфавита элементарных функций используется комбинированное множество функций, в которое вводится в том числе и множество аргументов математического выражения

$$\mathrm{F} = \{f_1 = q_1, \dots, f_p = q_p, f_{p+1} = x_1, \dots, f_{p+n} = x_n, \quad (3.123)$$
$$f_{m_0+1} = f_{1,1}(z), \dots, f_{m_0+m_1} = f_{1,m_1}(z),$$
$$f_{m_0+m_1+1} = f_{2,1}(z_1, z_2), \dots, f_{m_0+m_1+m_2} = f_{2,m_2}(z_1, z_2),$$
$$\dots$$
$$f_{K+1} = f_{r,1}(z_1, \dots, z_r), \dots, f_{K+m_r} = f_{r,m_r}(z_1, \dots, z_r)\},$$

где $m_0 = p + n$, $K = \sum_{i=0}^{r-1} m_i$.

Математическое выражение кодируется последовательностью номеров элементов из объединенного множества (3.123)

$$\mathrm{C} = (c_1, \dots, c_L), \quad (3.124)$$

где c_j – номер элемента из объединенного множества (3.123), $j = 1, \dots, L$.

В аналитическом программировании используется префиксный порядок элементов. Код функции в записи предшествует коду аргумента. Длина записи кода ограничена.

Рассмотрим пример.

Предположим, что определено следующее объединенное множество функций:

$$\mathrm{F} = \{q_1, q_2, x_1, x_2, -z, \sin(z), \cos(z), e^z, z_1 + z_1, z_1 z_2\} \quad (3.125)$$

и задан код математического выражения:

$$\mathrm{C}_1 = (10,9,10,1,7,3,10,2,6,4,8,5,3). \quad (3.126)$$

Декодирование заданного кода (3.126) осуществляется следующим образом.

Первый элемент $c_1^1 = 10$ соответствует функции умножения:

$$y_1 = z_1 z_2.$$

Следующий элемент $c_1^2 = 9$ соответствует функции сложения:

$$y_1 = (z_1 + z_3) z_2.$$

Затем $c_1^3 = 10$ – это снова умножение:

$$y_1 = (z_1 z_4 + z_3) z_2.$$

Далее $c_1^4 = 1$ – это параметр q_1:

$$y_1 = (q_1 z_4 + z_3) z_2.$$

$c_1^5 = 7$ – это косинус:

$$y_1 = (q_1 \cos(z_1) + z_3) z_2.$$

$c_1^6 = 3$ – это переменная x_1:

$$y_1 = (q_1 \cos(x_1) + z_3) z_2.$$

$c_1^7 = 10$ – снова умножение:

$$y_1 = (q_1 \cos(x_1) + z_1 z_3) z_2.$$

$c_1^8 = 2$ – это параметр q_2:

$$y_1 = (q_1 \cos(x_1) + q_2 z_3) z_2.$$

$c_1^9 = 6$ – это синус:

$$y_1 = \big(q_1 \cos(x_1) + q_2 \sin(z_1)\big) z_2 \, .$$

$c_1^{10} = 4$ – это переменная x_2:

$$y_1 = \big(q_1 \cos(x_1) + q_2 \sin(x_2)\big) z_2.$$

$c_1^{11} = 8$ – это e^z:

$$y_1 = (q_1 \cos(x_1) + q_2 \sin(x_2)) e^{z_1}.$$

$c_1^{12} = 5$ соответствует функции изменения знака:

$$y_1 = (q_1 \cos(x_1) + q_2 \sin(x_2)) e^{-z_1}.$$

$c_1^{13} = 3$ – это переменная x_1:

$$y_1 = (q_1 \cos(x_1) + q_2 \sin(x_2)) e^{-x_1}.$$

Корректность кода определяется индексом элемента кода. Индекс элемента i определяется из соотношения

$$T(i) = 1 - i + \sum_{j=1}^{i} a_j, \quad (3.127)$$

где

$$a_j = \begin{cases} 0, \text{если } a_j \leq m_0, \\ k, \text{если } \sum_{s=0}^{k} m_s < a_j \leq \sum_{s=0}^{k+1} m_s, \quad k = 0, \dots, r-1. \end{cases}$$

Для кодирования малых вариаций используется вектор из трех компонент:

$$\mathbf{w} = [w_1\ w_2\ w_3]^T,$$

где w_1 – тип малой вариации, w_2 – номер позиции кода, w_3 – новое значение кода.

Для данной кодировки определены следующие вариации

$w_1 = 1$ изменение номера функции с сохранением количества аргументов;

$w_1 = 2$ удаление номера функции из кода, если это функция с одним аргументом;

$w_1 = 3$ добавление функции с одним аргументом;

$w_1 = 4$ изменение номера функции так, чтобы количество аргументов функции увеличилось на единицу, при этом после этой функции добавляется код элемента из набора аргументов;

$w_1 = 5$ изменение номера функции так, чтобы количество аргументов функции уменьшилось на единицу, при этом из кода был удален первый код аргумента, который был найден после кода измененной функции.

Рассмотрим пример.

Пусть задан код аналитического программирования математического выражения (3.128) и объединенное множество элементарных функций (3.125).

Пусть задано следующее множество векторов вариаций:

$$\mathrm{W}_1 = \left(\begin{bmatrix} 5 \\ 1 \\ 6 \end{bmatrix}, \begin{bmatrix} 3 \\ 5 \\ 6 \end{bmatrix}, \begin{bmatrix} 3 \\ 10 \\ 7 \end{bmatrix}, \begin{bmatrix} 4 \\ 5 \\ 1 \end{bmatrix} \right) \quad (3.128)$$

Применим вариации к коду (3.128):

$$\mathbf{w}^1 \circ \mathrm{C} = (6,9,10,7,3,10,2,6,4,8,5,3).$$
$$\mathbf{w}^2 \circ \mathrm{C} = (6,9,10,7,6,3,10,2,6,4,8,5,3).$$
$$\mathbf{w}^3 \circ \mathrm{C} = (6,9,10,7,6,3,10,2,6,7,4,8,5,3).$$
$$\mathbf{w}^4 \circ \mathrm{C} = (6,9,10,7,9,1,3,10,2,6,7,4,8,5,3). \quad (3.129)$$

Новый полученный код (3.129) описывает следующее математическое выражение:

$$y = \sin(\cos(q_1 + x_1)q_2\sin(\cos(x_2)) + e^{-x_1}).$$

Метод вариационного аналитического программирования также был программно реализован, на него получено свидетельство о регистрации программ [212]. Он успешно применен во работах автора для синтеза системы стабилизации в задаче управления квадрокоптером [213].

3.7.4. Метод вариационного полного бинарного генетического программирования

Бинарное полное генетическое программирование (англ. Binary complete genetic programming BCGP) [214] также является разновидностью метода генетического программирования. BCGP представляет математическое выражение в виде бинарного дерева и использует только функции с одним и двумя аргументами. В вычислительном дереве функции с двумя аргументами связаны с узлами дерева. Функции с одним аргументом связаны с дугами.

Полное бинарное дерево имеет определенное количество элементов, зависящее от количества уровней. Каждый уровень, кроме последнего, содержит одинаковое количество функций с одним и двумя аргументами. Последний уровень содержит такое же количество функций с одним аргументом и аргументов математического выражения, которые схематически связаны с листьями дерева.

В BCGP математическое выражение представляет собой последовательную запись кодов функций и аргументов. Все коды расположены в определенном порядке по уровням. На каждом уровне сначала записываются коды функций с одним аргументом, затем функции с двумя аргументами и на

последнем уровне записываются коды аргументов математического выражения. Количество аргументов функции определяется порядковым номером элемента в коде.

Представление математического выражения в виде кода полного бинарного генетического программирования позволяет избежать дополнительных вычислительно затратных проверок корректности кода после выполнения генетических операций.

Определим алфавит элементарных функций:

- множество функций с двумя аргументами

$$\mathrm{F}_2 = \{f_1 = f_{2,1}(z_1, z_2), \dots, f_v = f_{2,v}(z_1, z_2)\}, \quad (3.130)$$

- множество функций с одним аргументом

$$G_1 = \{g_1 = f_{1,1}(z), \dots, g_w = f_{1,w}(z)\}, \quad (3.131)$$

- и множество аргументов математического выражения

$$\mathrm{A} = \{a_1 = q_1, \dots, a_p = q_p,$$

$$a_{p+1} = x_1, \dots, a_{n+p} = x_n, a_{n+p+1} = e_1, \dots, a_{n+p+v} = e_v\}, \quad (3.132)$$

где e_i – единичный элемент функции с двумя аргументами $f_{2,i}(z_1, z_2)$, $i = 1, \dots, v$.

Как видно, единичные элементы для функций с двумя аргументами добавляются ко множеству аргументов. Единичный элемент функции с двумя аргументами равен такому значению одного из аргументов, что результат вычисления функции равен значению другого аргумента. Единичным элементом для функции сложения является 0, а для функции умножения – 1.

Функции с двумя аргументами коммутативны:

$$f_{2,i}(z_1, z_2) = f_{2,i}(z_2, z_1), \ i = 1, \dots, v. \quad (3.133)$$

Начальным уровнем бинарного дерева является нулевой уровень. Пусть в бинарном дереве есть L уровней. На уровне $k < L$ бинарного дерева имеется 2^k функций с одним аргументом и столько же функций с двумя аргументами. На уровне $k = L$ находится 2^L функций с одним аргументом и такое же число

аргументов математического выражения с учетом единичных элементов для функций с двумя аргументами. Всего

$$N = 2\sum_{k=0}^{L} 2^k = 2(2^{L+1} - 1)$$

элементов в коде BCGP с L уровнями.

Рассмотрим пример.

Пусть задано математическое выражение:

$$y_1 = (x_1^2 - x_2^2)\cos(q_1 x_1 + q_2) + x_1 x_2 e^{-q_3 x_1}. \quad (3.134)$$

Для кодирования этого математического выражения достаточно следующих множеств функций и аргументов

$$\mathrm{F}_2 = \{f_1 = z_1 + z_2, f_2 = z_1 z_2\}, \quad (3.135)$$

$$G_1 = \{g_1 = z, g_2 = -z, g_3 = \cos(z), g_4 = e^z, g_5 = z^2\}, \quad (3.136)$$

$$\mathrm{A} = \{a_1 = q_1, a_2 = q_2, a_3 = q_3, a_4 = x_1, a_5 = x_2, a_6 = 0, a_7 = 1\}. \quad (3.137)$$

Бинарное дерево вычислений для математического выражения (3.134) показано на рис. 3.5.

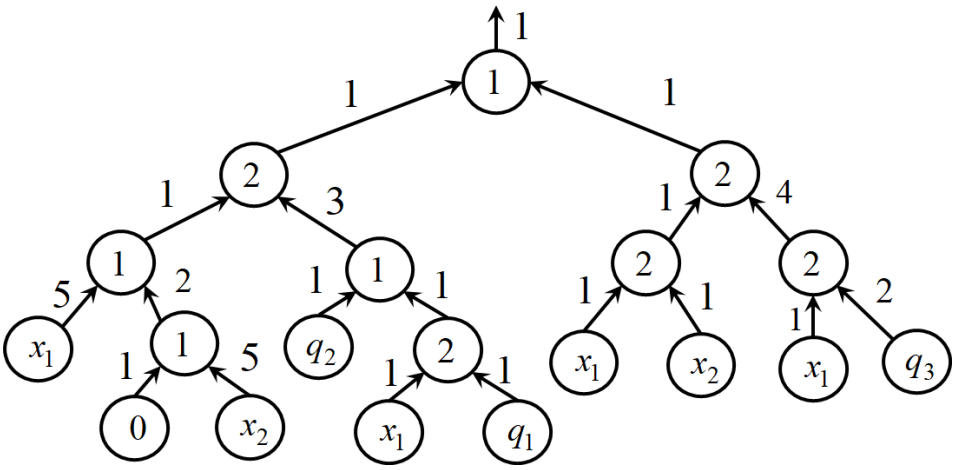

Рис.3.5. Бинарное дерево вычислений математического выражения (3.134)

Полученное бинарное дерево (рис.3.5) не является полным. Добавим в дерево дополнительные ветви и узлы с функциями с двумя аргументами и одиночными элементами этих функций, чтобы получить полное бинарное дерево.

Полное бинарное дерево вычислений для математического выражения (3.134) показано на рис.3.6.

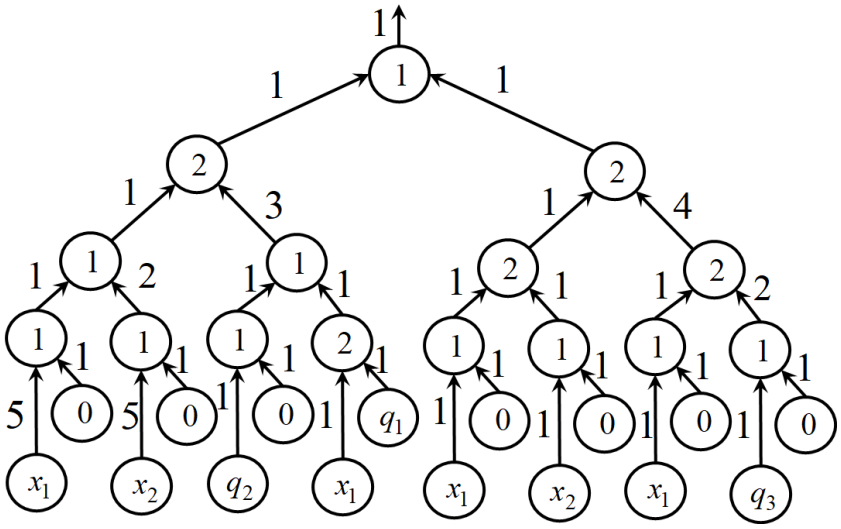

Рис.3.6. Полное бинарное дерево вычислений математического выражения (3.134) метода BCGP

Код BCGP представляет собой упорядоченное множество целых чисел, обозначающих номера элементов из множеств функций с одним и двумя аргументами, и аргументов закодированного математического выражения.

Код BCGP выражения (3.134) выглядит следующим образом:

$$\begin{aligned} C_0 = (&1,1,\\ &1,1,2,2,\\ &1,3,1,4,1,1,2,2,\\ &1,2,1,1,1,1,1,2,1,1,1,2,1,1,1,1,\\ &5,1,5,1,1,1,1,1,1,1,1,1,1,1,1,1,\\ &4,6,5,6,2,6,4,1,4,6,5,6,4,6,3,6). \end{aligned} \quad (3.138)$$

При генерации новых кодов математических выражений необходимо учитывать количество уровней бинарного дерева и генерировать коды математических выражений по структуре полного бинарного дерева. В результате коды всех математических выражений в процессе поиска имеют одинаковую длину. Во всех кодах аргументы математического выражения и функций с одним и двумя аргументами находятся в определенных позициях кода.

В бинарном полном генетическом программировании все коды функций с двумя аргументами, одним аргументом и аргументы математического выражения, в том числе и единичные элементы функций с двумя аргументами, в зависимости от количества уровней бинарного дерева всегда располагаются в определенных места. Поэтому малые вариации кода не меняют количество аргументов функции и не меняют длину кода.

Вектор малой вариации содержит только две компоненты:

$$\mathbf{w} = [w_1\ w_2]^T, \tag{3.139}$$

где w_1 – номер позиции, w_2 – новое значение элемента кода в соответствии с выбранной позицией.

Пусть бинарное дерево имеет L уровней. Для того, чтобы определить, к какому множеству принадлежит номер элемента, сначала определим уровень, на котором находится элемент w_1, по формуле:

$$2^l \le w_1 < 2^{l+1}. \tag{3.140}$$

Если найденный уровень не является последним $l < L$, то используется соотношение:

$$w_2 \in \begin{cases} \mathrm{G}_1, \text{если } w_1 \bmod 2^l \le 2^{l-1}, \\ \mathrm{F}_2, \text{иначе.} \end{cases} \tag{3.141}$$

Если $l = L$, то

$$w_2 \in \begin{cases} \mathrm{G}_1, \text{если } w_1 \bmod 2^l \le 2^{L-1}, \\ A, \text{иначе.} \end{cases} \tag{3.142}$$

Пусть $w_1 = 1$, тогда уровень $l = 0, 2^0 = 1 \le w_1 = 1 < 2^1 - 2$.

Тогда $r = w_1 \bmod 2^l = 0 \le 2^{-1} = 0{,}5$, поэтому $w_2 \in \mathrm{G}_1$.

Рассмотрим пример.

Пусть задан полный бинарный код генетического программирования (3.138) математического выражения (3.134) и алфавит элементарных функций (3.135)-(3.137).

Введем следующий вектор вариаций:

$$\mathrm{W}_1 = \left(\begin{bmatrix}7\\5\end{bmatrix}, \begin{bmatrix}32\\4\end{bmatrix}, \begin{bmatrix}11\\2\end{bmatrix}, \begin{bmatrix}48\\5\end{bmatrix}\right) \tag{3.143}$$

В результате применения данного вектора вариаций к коду базисного математического выражения (3.134) получаем новый BCGP код:

$$\mathbf{w}^4 \circ \mathbf{w}^3 \circ \mathbf{w}^2 \circ \mathbf{w}^1 \circ \mathrm{C}_0 = (1,1,$$
$$1,1,2,2,$$
$$5,3,1,4,1,1,2,2,$$
$$1,2,1,1,1,1,1,2,1,1,1,2,$$
$$1,1,1,1,$$
$$5,4,5,1,1,1,1,1,1,1,1,1,$$
$$1,1,1,1,4,5,5,6,2,6,4,$$
$$1,4,6,5,6,4,6,3,6). \quad (3.144)$$

Полученный код (3.144) соответствует следующему математическому выражению:

$$y_1 = (x_1^2 + e^{x_2})\cos(q_1 x_1 + q_2) + x_1 x_2 e^{-q_3 x_1}.$$

Данный метод программно реализован [216] и его применение к решению задачи синтеза представлено в работах автора [214, 215].

3.7.5. Метод вариационного Декартова генетического программирования

Декартово генетическое программирование (Cartesian Genetic Programming CGP) [217, 218] кодирует математическое выражение в виде набора целочисленных векторов. Каждый вектор содержит все необходимые коды для расчетов – это коды функции, ее аргументов и код переменной, куда следует записать результат расчета.

$$\mathrm{G} = (\mathbf{g}^1, \dots, \mathbf{g}^M), \quad (3.145)$$

где

$$\mathbf{g}^i = \left[g_1^i \dots g_R^i\right]^T, \quad (3.146)$$

g_1^i – номер функции, g_j^i – номер аргумента, $j = 2, \dots, R$, $i = 1, \dots, M$.

Для кодирования математического выражения необходимо определить базовый набор элементарных функций и набор аргументов математического выражения.

Пусть базовый набор включает в себя k_1 функций с одним аргументом, k_2 функций с двумя аргументами и k_3 функций с тремя аргументами. Тогда основной набор элементарных функций можно записать следующим образом:

$$\mathrm{F} = (f_1(z), \dots, f_{k_1}(z), f_{k_1+1}(z_1, z_2), \dots, f_{k_1+k_2}(z_1, z_2), \dots$$
$$f_{k_1+k_2+1}(z_1, z_2, z_3), \dots, f_{k_1+k_2+k_3}(z_1, z_2, z_3)). \qquad (3.147)$$

Определяем множество аргументов:

$$\mathrm{F}_0 = (x_1, \dots, x_n, q_1, \dots, q_p, e_1, \dots, e_{k_2}), \qquad (3.148)$$

где x_i – переменные, $i = 1, \dots, n$, q_j – постоянные параметр, $j = 1, \dots, p$, e_k – единичные элементы функций с двумя аргументами,

$$f_{k_1+l}(z_1, e_l) = z_1 \text{ и } f_{k_1+l}(e_l, z_2) = z_2.$$

Функции с тремя аргументами нужны для кодирования оператора if, часто применяемого в адаптивных системах управления. Например:

$$f(z_1, z_2, z_3) = \begin{cases} z_2, \text{если } z_1 \le 0 \\ z_3, \quad \text{иначе.} \end{cases}$$

Если для кодирования некоторого математического выражения использовать только функции с не более чем тремя аргументами, то размерность вектора кода элементарной функции (3.146) $R = 4$. Первая компонента – это номер функции из набора (3.147), а остальные компоненты – это номера аргументов из набора (3.148) или номера векторов (3.146), которые уже вычислены. Например:

$$\mathbf{g}^i = \left[g_1^i \;\; g_2^i \;\; g_3^i \;\; g_4^i\right]^T, \qquad (3.149)$$

где g_1^i – номер функции, если $g_1^i \le k_1$, то это функция с одним аргументом $f_{g_1^i}(z)$, если $k_1 < g_1^i \le k_1 + k_2$, то это номер функции с двумя аргументами $f_{g_1^i}(z_1, z_2)$, если $k_1 + k_2 < g_1 \le k_1 + k_2 + k_3$, то это функция с тремя аргументами, $f_{g_1^i}(z_1, z_2, z_3)$,

g_2^i, g_3^i, g_4^i – номера аргументов, если $1 \le g_k^i \le n + p + k_2$, $k = 2,3,4$, то это элемент множества аргументов (3.148), если $g_k^i > n + p + k_2$, $k = 2,3,4$, то g_k^i должно быть не больше $n + p + k_2 + i - 1$, в этом случае аргументом является результат вычисления функции, вызываемой вектором $\mathbf{g}^r = [g_1^r \;\; g_2^r \;\; g_3^r \;\; g_4^r]^T$, где

$r = g_k^i - n - p - k_2,\ \ k = 2,3,4$. Если g_1^i – это номер функции с одним аргументом, то компоненты g_3^i и g_4^i не используются. Если g_1^i – номер функции с двумя аргументами, то компонент g_4^i не используется.

Значения всех компонент вектора (3.149) должны удовлетворять ограничениям:

$$g_1^i \in \{1, \dots, k_1 + k_2 + k_3\},$$
$$g_k^i \in \{1, \dots, n + p + k_2 + i - 1\},\ k = 2,3,4. \quad (3.150)$$

Для вычисления математического выражения по коду Декартова генетического программирования нужен вектор результатов

$$\mathbf{y} = [y_1 \dots y_M]^T, \quad (3.151)$$

где

$$y_i = \begin{cases} f_{g_1^i}(g_2^i), \text{если } g_1^i \le k_1, \\ f_{g_1^i}(g_2^i, g_3^i), \text{если } k_1 < g_1^i \le k_2, \\ f_{g_1^i}(g_2^i, g_3^i, g_4^i), \text{если } k_2 < g_1^i \le k_3, \end{cases} \quad i = 1, \dots, M. \quad (3.152)$$

Рассмотрим пример.

Пусть задано следующее математическое выражение:

$$y_1 = e^{q_1 x_1}(\sin(q_2 x_2) + \cos(q_3 x_3)). \quad (3.153)$$

Для этого примера алфавит базовых множеств будет следующим:

$$\mathrm{F} = (f_1(z) = z, f_2(z) = -z, f_3(z) = \cos(z), f_4(z) = \sin(z), f_5(z) = e^z,$$
$$f_6(z_1, z_2) = z_1 + z_2, f_7(z_1, z_2) = z_1 z_2), \quad (3.154)$$
$$\mathrm{F}_0 = (x_1, x_2, x_3, q_1, q_2, q_3), \quad (3.155)$$

Для того, чтобы записать код математического выражения $q_1 x_1$, находим функцию умножения в наборе функций (3.156). Эта функция имеет номер 7, $f_7(z_1, z_2) = z_1 z_2$. Затем находятся номера элементов в наборе аргументов (3.155). Параметр q_1 – это элемент номер 4, переменная x_1 – это элемент 1. Четвертый компонент в коде не используется, поэтому может принимать любое значение согласно ограничениям (3.150), например $g_4^1 = 2$. В результате код $q_1 x_1$ запишется $\mathbf{g}^1 = [7\ \ 4\ \ 1\ \ 2]^T$. Тогда код $q_2 x_2$ равен $\mathbf{g}^2 = [7\ \ 5\ \ 2\ \ 3]^T$, а код $q_3 x_3$ равен $\mathbf{g}^3 = [7\ \ 6\ 3\ \ 4]^T$.

Далее записываем код $e^{q_1 x_1}$. Номер функции e^z равен 5. Эта функция имеет один аргумент, он является результатом вычисления $\mathbf{g}^1$. Номер аргумента $|\mathrm{F}_0| + 1 = 6 + 1 = 7$. В результате получается следующий вектор $\mathbf{g}^4 = [5\ \ 7\ \ 5\ \ 6]^T$. Компоненты $g_3^4 = 5$ и $g_4^4 = 6$ не используются.

В результате, код математического выражения (3.153) имеет следующий вид

$$\mathrm{G}_0 = \left(\begin{bmatrix} 7 \\ 4 \\ 1 \\ 2 \end{bmatrix}, \begin{bmatrix} 7 \\ 5 \\ 2 \\ 3 \end{bmatrix}, \begin{bmatrix} 7 \\ 6 \\ 3 \\ 4 \end{bmatrix}, \begin{bmatrix} 5 \\ 7 \\ 5 \\ 6 \end{bmatrix}, \begin{bmatrix} 4 \\ 8 \\ 1 \\ 2 \end{bmatrix}, \begin{bmatrix} 3 \\ 9 \\ 3 \\ 4 \end{bmatrix}, \begin{bmatrix} 6 \\ 11 \\ 12 \\ 5 \end{bmatrix}, \begin{bmatrix} 7 \\ 10 \\ 13 \\ 6 \end{bmatrix} \right). \tag{3.156}$$

Заранее неизвестно, сколько векторов нужно использовать для кодирования желаемой функции. Длина кода предварительно установлена как $L = 8$. Как правило, длина кодов выбирается избыточной, а лишние элементы просто записываются, но не учитываются.

Значение математического выражения может быть вычислено непосредственно по его закодированной форме.

Для применения принципа малых вариаций для Декартова генетического программирования необходимо определить малые вариации кода. Полагаем, что малая вариация кода Декартова генетического программирования – это изменение одного элемента кода. Тогда вектор малых вариаций для Декартова генетического программирования состоит из трех компонент:

$$\mathbf{w} = [w_1\ w_2\ w_3]^T, \tag{3.157}$$

где w_1 – номер вектора вызова (или номер столбца в коде), w_2 – номер изменяемого элемента в векторе вызова (номер строки в столбце w_1), w_3 – новое значение компоненты вектора вызова. Если $w_2 = 1$, то новое значение компоненты указывает номер какой-либо элементарной функции, если $w_2 > 1$, то w_3 указывает номер элемента из множества аргументов.

Рассмотрим пример.

Пусть имеем математическое выражение (3.153) и его код (3.156).

Множество аргументов положим (3.155), а множество функций расширим за счет включении функции с тремя переменными:

$$\mathrm{F} = (f_1(z) = z, f_2(z) = -z, f_3(z) = \cos(z), f_4(z) = \sin(z), f_5(z) = e^z,$$
$$f_6(z_1, z_2) = z_1 + z_2, f_7(z_1, z_2) = z_1 z_2, f_8 = f_{3,1}(z_1, z_2, z_3)), \quad (3.158)$$

где

$$f_{3,1}(z_1, z_2, z_3) = \begin{cases} z_2, \text{если } z_1 \leq 0, \\ z_3, \text{иначе.} \end{cases}$$

Пусть задано следующее множество векторов вариаций:

$$\mathrm{W}_1 = \left(\begin{bmatrix} 8 \\ 1 \\ 8 \end{bmatrix}, \begin{bmatrix} 3 \\ 3 \\ 8 \end{bmatrix}, \begin{bmatrix} 6 \\ 2 \\ 10 \end{bmatrix}, \begin{bmatrix} 1 \\ 3 \\ 2 \end{bmatrix}, \begin{bmatrix} 4 \\ 1 \\ 3 \end{bmatrix} \right) \quad (3.159)$$

В результате применения малых вариаций (3.159) к коду (3.156), получаем следующий код математического выражения:

$$\mathbf{w}^5 \circ \mathbf{w}^4 \circ \mathbf{w}^3 \circ \mathbf{w}^2 \circ \mathbf{w}^1 \circ \mathrm{G}_0 =$$
$$= \left(\begin{bmatrix} 7 \\ 4 \\ 2 \\ 2 \end{bmatrix}, \begin{bmatrix} 7 \\ 5 \\ 2 \\ 3 \end{bmatrix}, \begin{bmatrix} 7 \\ 6 \\ 8 \\ 4 \end{bmatrix}, \begin{bmatrix} 3 \\ 7 \\ 5 \\ 6 \end{bmatrix}, \begin{bmatrix} 4 \\ 8 \\ 1 \\ 2 \end{bmatrix}, \begin{bmatrix} 3 \\ 10 \\ 3 \\ 4 \end{bmatrix}, \begin{bmatrix} 6 \\ 11 \\ 12 \\ 5 \end{bmatrix}, \begin{bmatrix} 8 \\ 10 \\ 13 \\ 6 \end{bmatrix} \right). \quad (3.160)$$

Математическое выражение, соответствующее новому коду (3.160) следующее:

$$y = \begin{cases} \sin(q_2 x_2) + \cos(e^{q_1 x_2}), \text{если } cos(q_1 x_2) \leq 0, \\ q_3, \text{иначе.} \end{cases} \quad (3.161)$$

Определив малые вариации, осуществляем далее поиск оптимального решения на пространстве малых вариаций базисного решения.

Метод Вариационного Декартова генетического программирования реализован программно [219] и был протестирован для различных задач синтеза управления и показал хорошие результаты, представленные в работах [184, 220].

3.7.6. Многослойный метод сетевого оператора

Естественным развитием методов символьной регрессии, как и глубокого обучения в искусственных нейронных сетях, является переход к многослойным конструкциям.

Появление многослойных нейронных сетей было вызвано необходимостью аппроксимации более сложных функций, в частности, необходимостью аппроксимации функции исключающее ИЛИ. Методы символьной регрессии могут непосредственно искать сложные функции, нелинейные, разрывные и т.п. Однако с увеличением размерности решаемых задач, при поиске решений в виде функций большой размерности, размерность кода также значительно увеличивается, поэтому перспективным направлением может быть использование многослойного подхода.

Все методы символьной регрессии могут быть многослойными. Результат вычисления в одном слое добавляется ко множеству аргументов других слоев. В многослойных методах символьной регрессии во избежание цикличности необходимо устанавливать порядок слоев и контролировать вызовы результатов расчета в слоях. Главное правило, позволяющее избежать циклических вычислений – обеспечить порядок использования результатов вычислений по слоям. Слой с более высоким номером может использовать результаты вычислений слоев с более низким номером.

В рамках диссертационного исследования был предложен и апробирован многослойный метод символьной регрессии на основе сетевого оператора. Рассмотрим этапы метода многослойного сетевого оператора [221 - 225].

Рассмотрим код многослойного сетевого оператора с N слоями:

$$\Psi = (\Psi^1, \dots, \Psi^N). \tag{3.162}$$

Пусть заданы упорядоченные множества аргументов математического выражения и функций с одним и двумя аргументами.

$$\mathrm{F}_0 = \{x_1, \dots, x_n, q_1, \dots, q_p\}. \tag{3.163}$$

$$\mathrm{F}_1 = \{f_{1,1} = z, f_{1,2}(z), \dots, f_{1,w}(z)\}, \tag{3.164}$$

$$\mathrm{F}_2 = \{f_{2,1}(z_1, z_2), \dots, f_{2,v}(z_1, z_2)\}. \tag{3.165}$$

Предположим, что каждый сетевой оператор из (3.162) имеет разное количество входов и выходов. Определим целочисленные векторы соответствующих размерностей.

Для описания входов сетевого оператора вводится вектор входов:

$$\mathbf{r}^i = \left[r_1^i \dots r_{m_i}^i\right]^T, \tag{3.166}$$

где i – номер сетевого оператора, m_i – количество входных узлов сетевого оператора, $i = 1, \dots, N$.

Для описания выходов сетевого оператора используется выходной вектор

$$\mathbf{d}^i = \left[d_1^i \dots d_{n_i}^i\right]^T, \tag{3.167}$$

где i – номер сетевого оператора, n_i – количество узлов сетевого оператора, в которых хранятся результаты вычислений, d_j^i – номер узла в сетевом операторе Ψ^i, хранящий результаты вычислений, $j = 1, \dots, n_i$, $i = 1, \dots, N$, множество выходных узлов должно включать в себя номера всех выходных узлов сетевого оператора.

Во множество аргументов сетевого оператора на каждом слое входят результаты вычислений на предыдущих слоях.

Введем упорядоченное множество результатов вычислений сетевого оператора:

$$\mathrm{Y}_i = (y_1^i, \dots, y_{n_i}^i), \tag{3.168}$$

где y_j^i – результат вычислений сетевого оператора Ψ^i, хранящийся в узле с номером $d_j^i, j = 1, \dots, n_i, i = 1, \dots, N$.

Предположим, что номера входных узлов-источников r_j^i – это первые номера узлов сетевого оператора Ψ^i. Это всегда можно сделать, так как узлы-источники напрямую друг с другом не связаны.

Множество аргументов для каждого сетевого оператора разное.

Для первого сетевого оператора набор аргументов совпадает с набором аргументов (3.163) кодируемого математического выражения:

$$\mathrm{F}_0^1 = \mathrm{F}_0.$$

Для каждого следующего сетевого оператора во множество аргументов входят результаты расчетов предыдущих сетевых операторов.

$$\mathrm{F}_0^2 = \mathrm{F}_0^1 \cup \mathrm{Y}_1,$$
$$\mathrm{F}_0^3 = \mathrm{F}_0^2 \cup \mathrm{Y}_2,$$
$$\ldots$$
$$\mathrm{F}_0^N = \mathrm{F}_0^{N-1} \cup \mathrm{Y}_{N-1}.$$

Рассмотрим пример.

Пусть заданы множества аргументов математического выражения и функций с одним и двумя аргументами:

$$\mathrm{F}_0 = \{x_1, x_2, x_3, q_1, q_2, q_3\}, \tag{3.169}$$

$$\mathrm{F}_1 = \{f_{1,1}(z) = z, f_{1,2}(z) = -z, f_{1,3}(z) = \sin(z), \quad f_{1,4}(z) = \cos(z),$$
$$f_{1,5}(z) = e^z,\ f_{1,6}(z) = \arctan(z), f_{1,7}(z) = \sqrt[3]{z}, f_{1,8}(z) = \tanh(z)\}, \tag{3.170}$$

$$\mathrm{F}_2 = \{f_{2,1}(z_1, z_2) = z_1 + z_2, f_{2,2}(z_1, z_2) = z_1\, z_2\}. \tag{3.171}$$

Заданы $N = 4$ сетевых оператора разных размерностей:

$$\Psi^1 = \begin{bmatrix} 0&0&0&0&1&0&0&6 \\ 0&0&0&0&2&0&0&0 \\ 0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&2&0&0 \\ 0&0&0&0&2&0&3&0 \\ 0&0&0&0&0&2&4&0 \\ 0&0&0&0&0&0&2&0 \\ 0&0&0&0&0&0&0&1 \end{bmatrix}, \tag{3.172}$$

$$\Psi^2 = \begin{bmatrix} 0&0&1&0&1&0 \\ 0&0&2&0&0&0 \\ 0&0&2&5&0&0 \\ 0&0&0&1&1&0 \\ 0&0&0&0&2&8 \\ 0&0&0&0&0&1 \end{bmatrix}, \tag{3.173}$$

$$\Psi^3 = \begin{bmatrix} 0&0&0&0&1&0&0 \\ 0&0&0&0&1&0&0 \\ 0&0&0&0&0&6&0 \\ 0&0&0&0&0&1&0 \\ 0&0&0&0&2&1&0 \\ 0&0&0&0&0&2&7 \\ 0&0&0&0&0&0&1 \end{bmatrix}, \tag{3.174}$$

$$\Psi^4 = \begin{bmatrix} 0\,0\,0\,0\,1\,0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,1\,0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,5\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,6\,0\,0\,0\,0 \\ 0\,0\,0\,0\,1\,0\,4\,0\,0\,0 \\ 0\,0\,0\,0\,0\,2\,1\,1\,0\,0 \\ 0\,0\,0\,0\,0\,0\,2\,1\,0\,0 \\ 0\,0\,0\,0\,0\,0\,0\,1\,1\,0 \\ 0\,0\,0\,0\,0\,0\,0\,0\,1\,1 \\ 0\,0\,0\,0\,0\,0\,0\,0\,0\,1 \end{bmatrix}. \quad (3.175)$$

Вектора входов и выходов для сетевых операторов заданы:

$$\mathbf{r}^1 = [1\ 3\ 2\ 4]^T,$$
$$\mathbf{r}^2 = [3\ 6]^T,$$
$$\mathbf{r}^3 = [1\ 2\ 7\ 9]^T,$$
$$\mathbf{r}^4 = [11\ \ 12\ \ 8\ \ 10]^T, \quad (3.176)$$
$$\mathbf{d}^1 = [7\ \ 8]^T,$$
$$\mathbf{d}^2 = [5\ 6]^T,$$
$$\mathbf{d}^3 = [6\ 7]^T,$$
$$\mathbf{d}^4 = [10]. \quad (3.177)$$

Из векторов выходов получаются множества аргументов для каждого сетевого оператора.

$$\mathrm{F}_0^1 = (x_1, x_2, x_3, q_1, q_2, q_3), \quad (3.178)$$
$$\mathrm{F}_0^2 = (x_1, x_2, x_3, q_1, q_2, q_3, y_1^1, y_2^1), \quad (3.179)$$
$$\mathrm{F}_0^3 = (x_1, x_2, x_3, q_1, q_2, q_3, y_1^1, y_2^1, y_1^2, y_2^2), \quad (3.180)$$
$$\mathrm{F}_0^4 = (x_1, x_2, x_3, q_1, q_2, q_3, y_1^1, y_2^1, y_1^2, y_2^2, y_1^3, y_2^3). \quad (3.181)$$

Теперь можно получить математическое выражение по коду многослойного сетевого оператора.

По вектору входов (3.178) элементы множества аргументов x_1, x_2, q_1, q_2 подаются первому сетевому оператору в узлы $1, 2, 3, 4$ соответственно. В результате расчетов получаются следующие выражения:

$$y_1^1 = \sin(-q_1 x_1)\cos(-q_2 x_2),$$
$$y_2^1 = \sin(-q_1 x_1)\cos(-q_2 x_2) + \arctan(x_1).$$

Согласно вектору входов (3.179) для второго сетевого оператора на вход подаются x_3, q_3. Тогда:

$$y_1^2 = x_3 e^{-q_3 x_3},$$
$$y_2^2 = \tanh(x_3 e^{-q_3 x_3}).$$

Согласно вектору входов (3.180) для третьего сетевого оператора на вход подаются x_1, x_2, y_1^1 и y_1^2. Тогда:

$$y_1^3 = x_1 x_2 y_1^2 \arctan(y_1^1),$$
$$y_2^3 = \sqrt[3]{x_1 x_2 y_1^2 \arctan(y_1^1)}.$$

По входному вектору (3.181) на вход четвертого сетевого оператора подаются y_1^3, y_2^3, y_2^1, y_2^2. В результате расчетов по матрице сетевого оператора получаем следующее математическое выражение

$$y^4 = \cos(y_1^3 + y_2^3) e^{y_2^1} \arctan(y_2^2).$$

В результате, математическое выражение, закодированное четырехслойным сетевым оператором, имеет следующий вид:

$$y = \cos\left(x_1 x_2 y_1^2 \arctan(y_1^1) + \sqrt[3]{x_1 x_2 y_1^2 \arctan\big(\sin(-q_1 x_1)\cos(-q_2 x_2)\big)}\right) \times$$
$$e^{\sin(-q_1 x_1)\cos(-q_2 x_2) + \arctan(x_1)} \times \arctan(\tanh(x_3 e^{-q_3 x_3})). \qquad (3.182)$$

Генетические операции для многослойных структур символьной регрессии легко реализовать на основе принципа малых вариаций.

При использовании принципа малых вариаций базисного решения для методов многослойной символьной регрессии номер варьируемого слоя добавляется в код вариации в качестве первого компонента вектора вариации. Остальные компоненты вектора вариаций сохраняют свои прежние значения.

3.8. Примеры решения задачи синтеза

Рассмотрим несколько примеров решения задачи синтеза управления как задачи машинного обучения на основе методов символьной регрессии. Машинное обучение без учителя для синтеза управления предполагает прямой поиск функции управления на основе минимизации критерия качества.

3.8.1. Синтез системы управления для тестовой задачи оптимального управления

В классической монографии по оптимальному управлению [22] решена общая задача синтеза управления объектом, описываемым системой линейных дифференциальных уравнений второго порядка. Для сравнения с полученным решением решим эту же задачу методом символьной регрессии.

Задана математическая модель объекта управления

$$\dot{x}_1 = x_2, \tag{3.183}$$
$$\dot{x}_2 = u,$$

Заданы ограничения на управление

$$-1 \le u \le 1,\ u \in \mathrm{U} = [-1; 1]. \tag{3.184}$$

Задано терминальное состояние

$$\mathbf{x}^f = \begin{bmatrix} x_1^f & x_2^f \end{bmatrix}^T = [0 \quad 0]^T. \tag{3.185}$$

Необходимо найти управление, которое переводит объект из любой точки области начальных условий

$$-2 \le x_1 \le 2, ; -1.5 \le x_2 \le 1.5, \qquad \mathbf{X}_0 = [-2; 2] \times [-1.5; 1.5], \tag{3.186}$$

в терминальную точку (3.185) за минимальное время

$$J = t_f \to \min. \tag{3.187}$$

В отличие от классической формулировки задачи [22], здесь для численного решения задачи область начальных условий ограничена (3.186) и не является всем пространством состояний $\mathbb{R}^2$.

Для решения задачи численным методом символьной регрессии, вместо области начальных условий зададим конечное множество начальных условий из двадцати точек

$$\begin{aligned} \widetilde{\mathrm{X}}_0 = \{&\mathbf{x}^{\{0,1\}} = [-2 \ \ -1.5]^T, \mathbf{x}^{\{0,2\}} = [-2 \ \ -0.5]^T, \mathbf{x}^{\{0,3\}} = [-2 \ \ 0.5]^T, \\ &\mathbf{x}^{\{0,4\}} = [-2 \ \ 1.5]^T, \mathbf{x}^{\{0,5\}} = [-1 \ \ -1.5]^T, \mathbf{x}^{\{0,6\}} = [-1 \ \ -0.5]^T, \\ &\mathbf{x}^{\{0,7\}} = [-1 \ \ 0.5]^T, \mathbf{x}^{\{0,8\}} = [-1 \ \ 1.5]^T, \mathbf{x}^{\{0,9\}} = [0 \ \ -1.5]^T, \\ &\mathbf{x}^{\{0,10\}} = [0 \ \ -0.5]^T, \mathbf{x}^{\{0,11\}} = [0 \ \ 0.5]^T, \mathbf{x}^{\{0,12\}} = [0 \ \ 1.5]^T, \end{aligned} \tag{3.188}$$

$$\mathbf{x}^{\{0,13\}} = [1 \quad -1.5]^T, \mathbf{x}^{\{0,14\}} = [1 \quad -0.5]^T, \mathbf{x}^{\{0,15\}} = [1 \quad 0.5]^T,$$
$$\mathbf{x}^{\{0,16\}} = [1 \quad 1.5]^T, \mathbf{x}^{\{0,17\}} = [2 \quad -1.5]^T, \mathbf{x}^{\{0,18\}} = [2 \quad -0.5]^T,$$
$$\mathbf{x}^{\{0,19\}} = [2 \quad 0.5]^T, \mathbf{x}^{\{0,20\}} = [2 \quad 1.5]^{T\}}.$$

Управление ищем в виде функции координат пространства состояний

$$u = h\left(x_1^f - x_1, \;\; x_2^f - x_2\right) \in \mathrm{U}. \tag{3.189}$$

Переопределим функционал с учетом множества точек начальных условий (3.188)

$$\tilde{J} = \sum_{i=1}^{20} \left(t_{f,i} + p_1 \left\| \mathbf{x}^f - \mathbf{x}\left(t_{f,j}, \mathbf{x}^{0,i}\right) \right\|_2\right) \to \min, \tag{3.190}$$

где

$$t_{f,i} = \begin{cases} t, \;\; \text{если } \left\| \mathbf{x}\left(t, \mathbf{x}^{0,i}\right) - \mathbf{x}^f \right\|_2 \le \varepsilon_0, \\ t^+ - \text{иначе}, \end{cases} \tag{3.191}$$

$\mathbf{x}\left(t, \mathbf{x}^{0,i}\right)$ – частное решение системы (3.183) с управлением (3.189) из начальных условий $\mathbf{x}^{0,i}$, $i \in \{1, \dots, 20\}$, $t^+ = 5.1$ сек, $\varepsilon_0 = 0.01$, p_1 – весовой коэффициент, $p_1 = 1$,

$$\left\| \mathbf{x}\left(t, \mathbf{x}^{0,i}\right) - \mathbf{x}^f \right\|_2 = \sqrt{\sum_{i=1}^{2} \left(x_i\left(t, \mathbf{x}^{0,i}\right) - x_i^f\right)^2}. \tag{3.192}$$

Для решения задачи был применен численный метод сетевого оператора. Размерность матрицы сетевого оператора составляла 14×14. базисное решение было выбрано в виде пропорционального регулятора:

$$u = \begin{cases} 1, \text{если } q_1\left(x_1^f - x_1\right) + q_2\left(x_2^f - x_2\right) \ge \; 1, \\ -1, \text{если } q_1\left(x_1^f - x_1\right) + q_2\left(x_2^f - x_2\right) \le \; -1, \\ q_1\left(x_1^f - x_1\right) + q_2\left(x_2^f - x_2\right) - \text{иначе}, \end{cases} \tag{3.193}$$

где $q_1 = 1$, $q_2 = 1$.

В результате машиной было получено следующее решение

$$u = \begin{cases} 1, \text{если } \tilde{u} \ge \; 1, \\ -1, \;\; \text{если } \tilde{u} \le \; -1, \\ \tilde{u} - \text{иначе}, \end{cases} \tag{3.194}$$

где

$$\tilde{u} = \arctan(A) + \frac{1}{B} + C^3, \tag{3.195}$$

$$A = B + \arctan\left(q_1 \mathrm{sgn}\left(x_1^f - x_1\right)\sqrt{\left|x_1^f - x_1\right|}\right),$$

$$B = q_1(C + \mathrm{sgn}(C)\exp(-|C|)),$$

$$C = q_2\left(x_2^f - x_2\right) + \mathrm{sgn}\left(x_1^f - x_1\right)\sqrt{\left|x_1^f - x_1\right|},$$

$$q_1 = 11.33423,\ q_2 = 8.13892.$$

На рис.3.7 приведены результаты моделирования системы (3.189) с функцией управления (3.194)- (3.195), синтезированной машинным обучением с помощью метода сетевого оператора. На графике представлены фазовые траектории из восьми начальных условий:

$\mathbf{x}^{0,1} = [-2 \quad -1.5]^T$, $\mathbf{x}^{0,4} = [-2 \quad 1.5]^T$, $\mathbf{x}^{0,9} = [0 \quad -1.5]^T$, $\mathbf{x}^{0,12} = [0 \quad 1.5]^T$, $\mathbf{x}^{0,17} = [2 \quad -1.5]^T$, $\mathbf{x}^{0,20} = [2 \quad 1.5]^T$, $\mathbf{x}^{0,21} = [-2 \quad 0]^T$, $\mathbf{x}^{0,22} = [2 \quad 0]^T$.

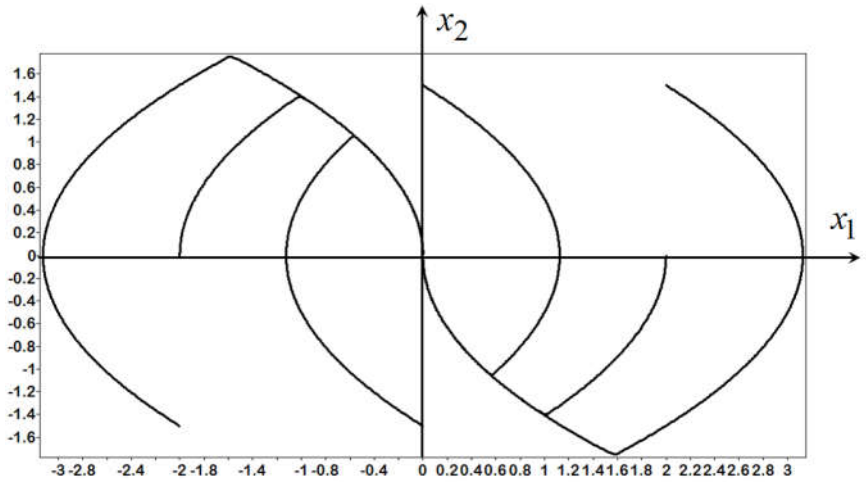

Рис.3.7. Фазовые траектории объекта из восьми начальных условий для функции управления (3.194)- (3.195), полученной машинным обучением с помощью метода сетевого оператора

Для сравнения полученных результатов с классическим результатом на рис. 3.8 приведены результаты моделирования системы (3.189) из тех же начальных условий с оптимальным управлением, полученным с помощью принципа максимума Понтрягина

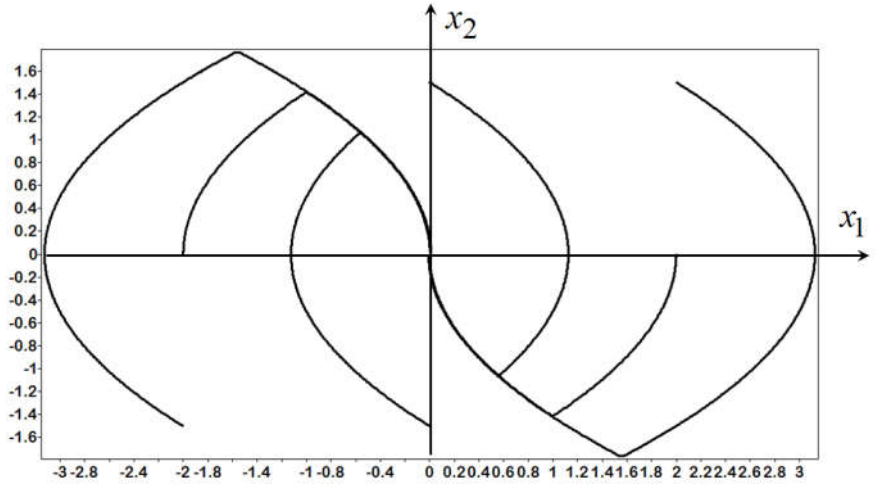

Рис.3.8. Оптимальные траектории объекта из восьми начальных условий для функции управления, полученной с помощью принципа максимума

Все траектории на рис. 3.7 и рис.3.8 совпадают с высокой точностью, поэтому они и не представлены на одном графике.

Далее на рис 3.9 приведены результаты моделирования системы (3.189) с машинно обученным управлением (3.194)- (3.195) из начальных условий, не входивших в заданное множество (3.188), которое использовалось при решении задачи синтеза управления.

Моделирование проводилось для следующих начальных условий: $\mathbf{x}^{0,23} = [-1.5 \;\; -1]^T$, $\mathbf{x}^{0,24} = [-0.5 \;\; -1]^T$, $\mathbf{x}^{0,25} = [0.5 \;\; -1]^T$, $\mathbf{x}^{0,26} = [1.5 \;\; -1]^T$, $\mathbf{x}^{0,27} = [-1.5 \;\; 1]^T$, $\mathbf{x}^{0,28} = [-0.5 \;\; 1]^T$, $\mathbf{x}^{0,29} = [0.5 \;\; 1]^T$, $\mathbf{x}^{0,30} = [1.5 \;\; 1]^T$.

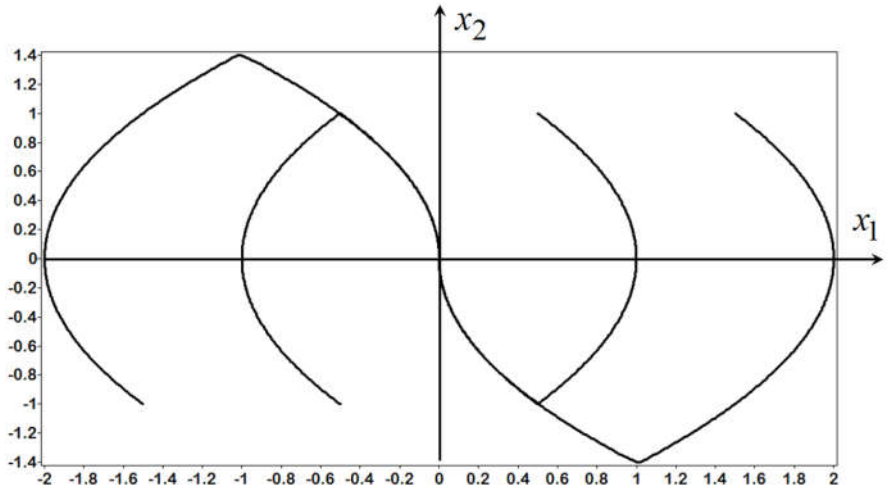

Рис.3.9. Фазовые траектории объекта при машиннно синтезированном управлении (3.194)- (3.195) из восьми начальных условий, не входивших в обучающее множество

Для сравнения на рис. 3.10 приведены результаты моделирования системы из тех же начальных условий при оптимальном управлении, полученном с помощью принципа максимума.

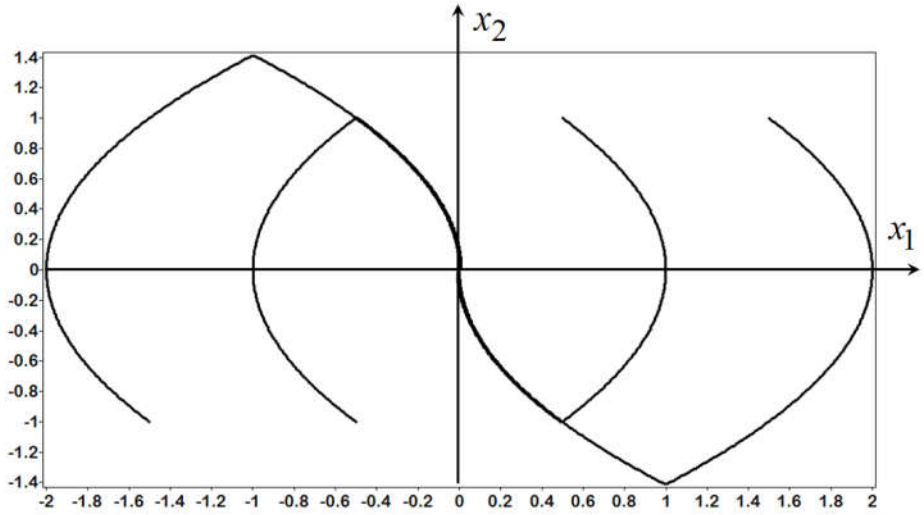

Рис.3.10. Оптимальные траектории объекта из восьми начальных условий, не входящих в начальное множество (3.188) для функции управления, полученной с помощью принципа максимума

Как видим из сравнения графиков, оптимальные траектории, полученные с помощью принципа максимума и траектории, полученные при машинно синтезированном управлении совпадают с высокой точностью. Значения функционала для всех начальных условий для оптимального и синтезированного управлений совпали с точность до 0.01 сек. Из результатов моделирования можно сделать вывод, что метод сетевого оператора синтезировал оптимальную систему управления.

3.8.2. Синтез системы стабилизации для мобильного робота

Далее рассмотрим более прикладную задачу пространственной стабилизации мобильного робота.

Простая математическая кинематическая модель робота [226] описывается следующей системой обыкновенных дифференциальных уравнений:

$$\dot{x}_1 = 0.5(u_1 + u_2)\cos(x_3),$$
$$\dot{x}_2 = 0.5(u_1 + u_2)\sin(x_3), \quad (3.196)$$
$$\dot{x}_3 = 0.5(u_1 - u_2),$$

где $\mathbf{x} = [x_1 \;\; x_2 \;\; x_3]^T$ – вектор состояния, $\mathbf{u} = [u_1 \;\; u_2]^T$ – вектор управления.

Значения элементов вектора управления ограничены:

$$-10 \leq u_i \leq 10, i = 1,2, \;\; \mathbf{u} \in \mathrm{U} = [-10 \;\; 10] \times [-10; 10]. \quad (3.197)$$

Для модели (3.196) задано целевое терминальное состояние

$$\mathbf{x}^f = [0 \;\; 0 \;\; 0]^T. \quad (3.198)$$

Необходимо найти управление в виде функции от координат пространства состояний:

$$\mathbf{u} = \mathbf{h}(\mathbf{x}^f - \mathbf{x}). \quad (3.199)$$

Функция (3.199) должна обеспечивать перемещение робота в целевую терминальную точку из любого начального состояния области:

$$\widetilde{\mathrm{X}}_0 = [-2 \;\; 2] \times [-2 \;\; 2] \times \left[-\frac{\pi}{4} \;\; \frac{\pi}{4}\right] \in \mathbb{R}^3, \quad (3.200)$$

с критерием быстродействия:

$$J = t_f \to \min. \quad (3.201)$$

Для численного решения задачи методом символьной регрессии диапазон начальных значений заменяется конечным набором точек начальных условий:

$$\mathrm{X}_0 = \{\mathbf{x}^{0,k} = \left[x_1^{0,k}(i_1) \;\; x_2^{0,k}(i_2) \;\; x_3^{0,k}(i_3)\right]^T:$$
$$k = i_1 + (i_2 - 1)i_1^+ + (i_3 - 1)i_1^+ i_2^+\}, \quad (3.202)$$

где

$$x_1^{0,k}(i_1) = x_1^+ - (i_1 - 1)\delta_1,$$
$$x_2^{0,k}(i_2) = x_2^+ - (i_2 - 1)\delta_2,$$
$$x_3^{0,k}(i_3) = x_3^+ - (i_3 - 1)\delta_3,$$

$i_1 = 1, \ldots, i_1^+$, $i_2 = 1, \ldots, i_2^+$, $i_3 = 1, \ldots, i_3^+$, $x_1^+ = 2$, $x_2^+ = 2$, $x_3^+ = \pi/4$, $\delta_1 = 1$, $\delta_2 = 1$, $\delta_3 = \pi/4$, $i_1^+ = 5$, $i_2^+ = 5$, $i_3^+ = 3$. Всего при машинном синтезе в настоящем эксперименте учитывалось $5 \cdot 5 \cdot 3 = 75$ начальных условий заданной области (3.200).

Соответственно внесем изменения и в функционал качества:

$$J_e = t_f + \frac{1}{72}\sum_{i=1}^{72}\sqrt{\sum_{j=1}^{3}\left(x_j\left(t_{f,i},\ \mathbf{x}^{0,i}\right) - x_j^f\right)^2} \to \min, \qquad (3.203)$$

где

$$t_f = \max\{\mathbf{x}\left(t_{f,i},\ \mathbf{x}^{0,i}\right):\quad i = 1,\dots,72\}, \qquad (3.204)$$

$t_{f,i}$ определяется по формуле (3.191), $\mathbf{x}\left(t_{f,i},\ \mathbf{x}^{0,i}\right)$ – частное решение системы (3.196) с управлением (3.199) из начального условия $\mathbf{x}^{0,i}$, $i \in \{1,\dots,30\}$, $t^+ = 1.5$ сек, $\varepsilon_0 = 0.01$.

Для демонстрации возможностей вычислительных методов символьной регрессии решим данную задачу несколькими разными методами.

Сначала применим метод Вариационного Декартова генетического программирования VCGP (см. раздел 3.7.5).

В качестве базисного решения был выбран пропорциональный регулятор:

$$u_1 = q_1(x_1^f - x_1) + q_2(x_2^f - x_2) + q_3(x_3^f - x_3),$$
$$u_2 = q_1\left(x_1^f - x_1\right) + q_2(x_2^f - x_2) + q_3(x_3^f - x_3) \qquad (3.205)$$

где $\mathbf{q} = [q_1\ \ q_2\ \ q_3]^T$ – вектор параметров.

Базовое множество было задано следующим образом:

$$\mathrm{F}_0 = \{x_1^f - x_1, x_2^f - x_2, x_3^f - x_3, q_1, q_2, q_3\}. \qquad (3.206)$$

Для поиска решения был определен код VCGP, длиной $M = 24$, в котором закодированы вызовы функций с одним, двумя и тремя аргументами, поэтому каждый вектор вызова состоял из 4 компонент.

Код VCGP базисного решения имел следующий вид:

$$\mathrm{G}(u_i) = \left(\begin{bmatrix}30\\1\\4\\1\end{bmatrix},\begin{bmatrix}30\\2\\5\\2\end{bmatrix},\begin{bmatrix}30\\3\\6\\3\end{bmatrix},\begin{bmatrix}29\\7\\8\\4\end{bmatrix},\begin{bmatrix}29\\9\\10\\5\end{bmatrix},\begin{bmatrix}1\\11\\6\\1\end{bmatrix},\begin{bmatrix}1\\12\\2\\3\end{bmatrix},\begin{bmatrix}1\\13\\4\\5\end{bmatrix},\right.$$
$$\left.\begin{bmatrix}1\\14\\1\\4\end{bmatrix},\begin{bmatrix}1\\15\\2\\5\end{bmatrix},\begin{bmatrix}1\\16\\3\\6\end{bmatrix},\begin{bmatrix}1\\17\\7\\8\end{bmatrix},\begin{bmatrix}1\\18\\9\\10\end{bmatrix},\begin{bmatrix}1\\19\\11\\12\end{bmatrix},\begin{bmatrix}1\\20\\13\\14\end{bmatrix},\begin{bmatrix}1\\21\\15\\16\end{bmatrix},\right.$$

$$\left.\begin{bmatrix}1\\22\\17\\18\end{bmatrix},\begin{bmatrix}1\\23\\18\\19\end{bmatrix},\begin{bmatrix}1\\24\\20\\21\end{bmatrix},\begin{bmatrix}1\\25\\21\\22\end{bmatrix},\begin{bmatrix}1\\26\\23\\24\end{bmatrix},\begin{bmatrix}1\\27\\21\\22\end{bmatrix},\begin{bmatrix}1\\28\\23\\24\end{bmatrix},\begin{bmatrix}1\\29\\25\\26\end{bmatrix}\right) \quad (3.207)$$

Для поиска использовались 40 машинно реализуемых функций, программное описание которых представлено подробно в приложении 1.

- $k_1 = 28$ функций с одним аргументом:

$$f_1(z) = z,\ f_2(z) = z^2,\ f_3(z) = -z,\ f_4(z) = \operatorname{sgn}(z)\sqrt{|z|},$$

$$f_5(z) = z^{-1},\ f_6(z) = \exp(z),\ f_7(z) = \log(|z|),\ f_8(z) = \tanh(0.5z),$$

$$f_9(z) = \begin{cases}1, \text{если } z \geq 0,\\ 0 - \text{иначе},\end{cases}\ f_{10}(z) = \operatorname{sgn}(z),\ f_{11}(z) = \cos(z),\ f_{12}(z) = \sin(z),$$

$$f_{13}(z) = \arctan(z),\ f_{14}(z) = z^3,\ f_{15}(z) = \sqrt[3]{z},\ f_{16}(z) = \begin{cases}z, \text{если } |z| < 1,\\ \operatorname{sgn}(z) - \text{иначе},\end{cases}$$

$$f_{17}(z) = \operatorname{sgn}(z)\log(|z| + 1),\ f_{18}(z) = \operatorname{sgn}(z)(\exp(z) - 1),$$

$$f_{19}(z) = \operatorname{sgn}(z)\exp(-|z|),\ f_{20}(z) = 0.5z,\ f_{21}(z) = 2z,$$

$$f_{22}(z) = 1 - \exp(-|z|),\ f_{23}(z) = z - z^3,\ f_{24}(z) = \frac{1}{1+\exp(-|z|)},$$

$$f_{25}(z) = \begin{cases}1, \text{если } z > 0,\\ 0 - \text{иначе},\end{cases}\ f_{26}(z) = \begin{cases}1, \text{если } |z| < \varepsilon,\\ \operatorname{sgn}(z) - \text{иначе},\end{cases}$$

$$f_{27}(z) = \operatorname{sgn}(z)\left(1 - \sqrt[2]{1 - z^2}\right),\ f_{28}(z) = z(1 - \exp(-z^2)),$$

- $k_2 = 8$ функций с двумя аргументами:

$$f_{29}(z_1, z_2) = z_1 + z_2,\ f_{30}(z_1, z_2) = z_1 z_2,\ f_{31}(z_1, z_2) = \max\{z_1, z_2\},$$

$$f_{32}(z_1, z_2) = \min\{z_1, z_2\},\ f_{33}(z_1, z_2) = z_1 + z_2 - z_1 z_2,$$

$$f_{34}(z_1, z_2) = \operatorname{sgn}(z_1 + z_2)\sqrt{z_1^2 + z_2^2},\ f_{35}(z_1, z_2) = \operatorname{sgn}(z_1 + z_2)(|z_1| + |z_2|),$$

$$f_{36}(z_1, z_2) = \operatorname{sgn}(z_1 + z_2)|z_1||z_2|,$$

- и $k_3 = 4$ функций с тремя аргументами:

$$f_{37}(z_1, z_2, z_3) = \begin{cases} z_2, \text{если } z_1 > 0, \\ \mathrm{z}_3 - \text{иначе}, \end{cases} \qquad f_{38}(z_1, z_2, z_3) = \begin{cases} z_3, \text{если } z_1 > z_2, \\ -\mathrm{z}_3 - \text{иначе}, \end{cases}$$

$$f_{39}(z_1, z_2\,, z_3) = \begin{cases} z_2 + z_3, \text{если } z_1 > 0, \\ z_2 - z_3 - \text{иначе}, \end{cases} \qquad f_{40}(z_1, z_2\,, z_3) = \max\{z_1, z_2, z_3\}.$$

В результате машина с помощью вариационного Декартова генетического программирования нашла следующую функцию управления:

$$u_i = \begin{cases} 10, \text{если } \tilde{u}_i \geq\ 10, \\ -10,\ \text{если } \tilde{u}_i \leq\ -10, \\ \tilde{u}_i - \text{иначе}, \end{cases} \qquad i = 1{,}2 \qquad (3.208)$$

где

$$\tilde{u}_1 = \mathrm{sgn}(q_3 A - 3q_2(x_3^f - x_3) + \ q_1(x_1^f - x_1)) - (x_3^f - x_3)) \times$$

$$(\exp|q_3 \arctan(3q_2 A - 3q_2(x_3^f - x_3) + \ q_1(x_1^f - x_1)) - (x_3^f - x_3)| - 1),$$

$$\tilde{u}_2 = \sqrt[3]{-2q_2 A + 2q_2\left(x_3^f - x_3\right) + \mathrm{sgn}\left(\ x_1^f - x_1\right)} - 2q_2 A + 2q_2\left(x_3^f - x_3\right) +$$

$\mathrm{sgn}(\ x_1^f - x_1) + \ q_1(x_1^f\ - x_1)$,

$$A = \arctan(q_1\left(x_1^f - x_1\right)q_2(x_2^f - x_2)),$$

$$q_1 = 3.61914,\ q_2 = 3.85645,\ q_3 = 3.36719.$$

На рис. 3.11 представлены результаты моделирования объекта управления (3.196) с найденной функцией управления (3.208) из восьми начальных условий:

$$\mathbf{x}^{0,26} = [2\ \ 2\ \ \ 0]^T,\ \mathbf{x}^{0,27} = [1\ \ 2\ \ 0]^T,$$

$$\mathbf{x}^{0,29} = [-1\ \ 2\ \ 0]^T,\ \mathbf{x}^{0,30} = [-2\ \ 2\ \ 0]^T,$$

$$\mathbf{x}^{0,46} = [2\ \ -2\ \ \ 0]^T,\ \mathbf{x}^{0,47} = [1\ \ -2\ \ \ 0]^T,$$

$$\mathbf{x}^{0,49} = [2\ \ -2\ \ 0]^T,\ \mathbf{x}^{0,50} = [-2\ \ -2\ \ \ 0]^T.$$

Как видно из рис.3.11, объект достигает конечного состояния из различных заданных начальных условий.

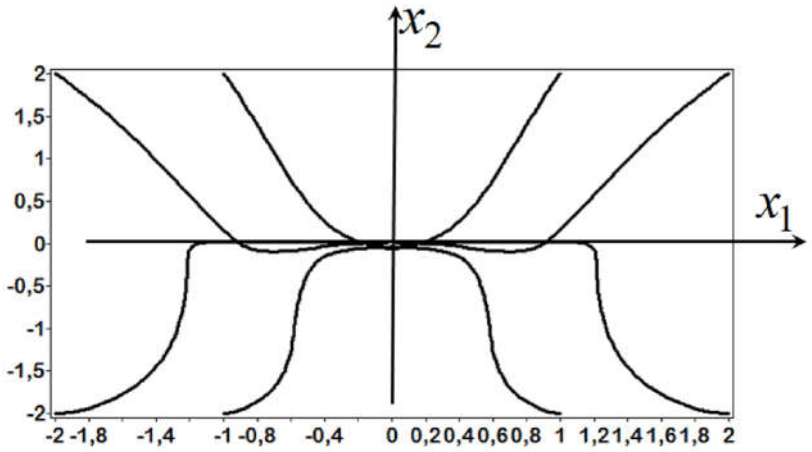

Рис.3.11. Траектории робота из 8 начальных условий с функцией управления, машинно синтезированной методом Вариационного Декартова генетического программирования

Та же задача управления была решена еще двумя методами символьной регрессии, методом сетевого оператора и вариационным полным бинарным генетическим программированием.

В качестве базисного решения во всех алгоритмах использовались пропорциональные регуляторы по каждой переменной. В качестве бинарных операций использовались операции сложения и умножения, а в качестве унарных – набор из 28 элементарных функций.

С помощью метода сетевого оператора (см. раздел 3.7.1.) был найден следующий закон управления:

$$u_i = \begin{cases} 10, \text{если } \tilde{u}_i \geq 10, \\ -10, \text{ если } \tilde{u}_i \leq -10, \\ \tilde{u}_i - \text{иначе}, \end{cases} \qquad i = 1,2 \qquad (3.209)$$

где

$$\tilde{u}_1 = A\ln(|tanh(C) + \mu(D) + F^3 + G + \sin(q_3(x_3^f - x_3))|),$$

$$\tilde{u}_2 = \tilde{u}_1 + \sin(A) + \mu(A) + B^{-1} + \operatorname{sgn}(C)\ln(|C| + 1) + \arctan(D) + \tanh(E)$$

$$+\mu\left(G + \tanh(H) + \left(x_1^f - x_1\right)\right) + G - G^3 + q_3^{-2}q_1^{-1}\left(x_1^f - x_1\right)^{-1} +$$

$$\operatorname{sgn}(x_1^f - x_1),$$

$$A = B^{-1} + \sqrt[3]{\tanh(C) + \mu(D) + F^3 + G + \sin\left(q_3\left(x_3^f - x_3\right)\right)} + \operatorname{sgn}(q_3(x_3^f - x_3))\exp(-|q_3(x_3^f - x_3)|),$$

$$B = \tanh(C) + \mu(D) + F^3 + 2G + \sin\left(q_3(x_3^f - x_3)\right) + \tanh(H) + (x_1^f - x_1) - \left(G + \tanh(H) + (x_1^f - x_1)\right)^3,$$

$$C = D + G - G^3 + \operatorname{sgn}(q_3^2 q_1(x_1^f - x_1)) + \arctan(q_1) + \vartheta(x_3^f - x_3),$$

$$D = E + \sqrt[3]{F} + \operatorname{sgn}\left(G + \tanh(H) + \left(x_1^f - x_1\right)\right) + \operatorname{sgn}(G)\sqrt{\{|G|\}} + \arctan(q_3^2 q_1\left(x_1^f - x_1\right)),$$

$$E = F + G + \tanh(H) + \left(x_1^f - x_1\right) + H + \operatorname{sgn}(q_3^2 q_1(x_1^f - x_1))\sqrt{|q_3^2 q_1(x_1^f - x_1)|},$$

$$F = G + \tanh(H) + \left(x_1^f - x_1\right) + \tanh(q_3^2 q_1\left(x_1^f - x_1\right)) + \sqrt[3]{\left(x_1^f - x_1\right)},$$

$$G = H + \operatorname{sgn}(q_2\left(x_2^f - x_2\right) + \operatorname{sgn}\left(x_1^f - x_1\right)) * \exp(-|q_2\left(x_2^f - x_2\right)\operatorname{sgn}\left(x_1^f - x_1\right)|) + \cos(q_3(x_3^f - x_3)) + \sin\left(x_1^f - x_1\right),$$

$$H = q_2\left(x_2^f - x_2\right) + \operatorname{sgn}\left(x_1^f - x_1\right) + q_3(x_3^f - x_3) + \tanh(q_3^2 q_1\left(x_1^f - x_1\right)),$$

$$\mu(\alpha) = \begin{cases} \alpha, \text{если } |\alpha| \leq 1, \\ \operatorname{sgn}(\alpha) - \text{иначе}, \end{cases}$$

$$q_1 = 15.80103,\ q_2 = 14.63843,\ q_3 = 13.00757.$$

Как можно заметить, выражение, полученное методом сетевого оператора, гораздо более массивное, нежели при вариационном Декартовом генетическом программировании. Однако, это лишь интерпретация полученных кодов в привычный для нас вид, а для машины, как бортового вычислителя и та и другая функции просто закодированы в виде массивов и вычисляются согласно заложенной стандартной процедуры.

На рис.3.12 представлены результаты моделирования системы (3.196) с функцией управления (3.209) из различных начальных условий.

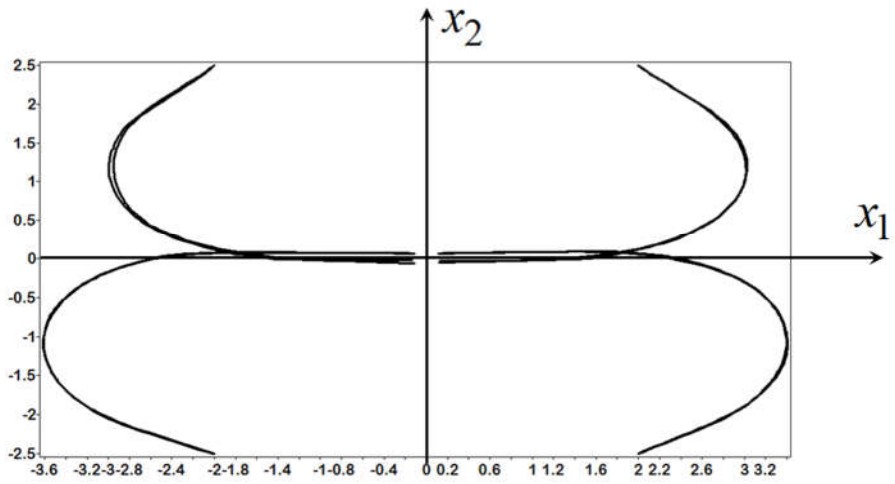

Рис.3.12. Траектории робота с функцией управления, машинно синтезированной методом сетевого оператора

Далее задача решалась с помощью метода вариационного полного бинарного генетического программирования (см. раздел 3.7.4) и было получено следующее решение:

$$u_i = \begin{cases} 10, \text{если } \tilde{u}_i \geq 10, \\ -10, \text{ если } \tilde{u}_i \leq -10, \\ \tilde{u}_i - \text{иначе}, \end{cases} \qquad i = 1,2 \tag{3.210}$$

где

$$\tilde{u}_1 = \left(q_2 + \left(x_2^f - x_2\right)\right)\left(x_1^f - x_1\right)\cos(x_3^f - x_3) + \sqrt[3]{q_3}\arctan(\sqrt[3]{(x_3^f - x_3)}) +$$

$$\left(\left(\sqrt[3]{q_3} + q_1\right)\left(x_2^f - x_2\right)\sin(x_3^f - x_3)(\,2q_2 - q_2^3)\right)^3,$$

$$\tilde{u}_2 = (2q_3 + 1)\left(\left(x_2^f - x_2\right)\left(x_1^f - x_1\right)^{-1} - \left(x_3^f - x_3\right)\right)^{-1} +$$

$$(q_3^3\left(x_2^f - x_2\right) + \exp(\left(x_1^f - x_1\right)))(\sin(\exp\left(x_2^f - x_2\right)\left(x_1^f - x_1\right)^{-1}) - (x_3^f - x_3)),$$

$$q_1 = 3.33594,\ q_2 = 3.77930,\ q_3 = 2.52148.$$

На рис.3.13 представлены результаты моделирования системы (3.196) с функцией управления (3.210) из различных начальных условий.

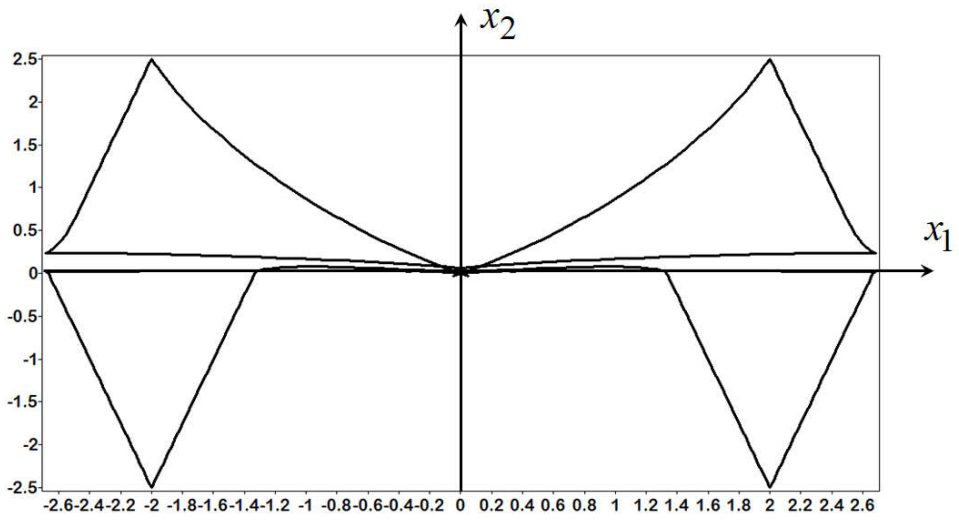

Рис.3.13. Траектории робота с функцией управления, машинно синтезированной методом вариационного полного бинарного генетического программирования

Как можно полагать по типу траектории движения объекта к цели, что само движение объекта не является оптимальным с точки зрения длины траектории. Эта ситуация распространена в машинном обучении без учителя, когда обучение системы стабилизации основывается только на значении функционала качества. В реальности компьютер интуитивно не знает, как двигаться к цели, поэтому машинный интеллект определил этот тип движения на основе минимизации функционала. При необходимости система может быть дообучена.

Целью экспериментов было показать, что вычислительные методы символьной регрессии позволяют получить функцию управления, которая при подстановке в правые части системы дифференциальных уравнений, описывающих модель объекта управления, делает этот объект устойчивым. В результате было продемонстрировано, что различные методы символьной регрессии позволяют успешно решать данную задачу машинного обучения с хорошим качеством без трудоемкого построения обучающей выборки, основываясь только на критерии минимизации функционала качества.

3.9. Выводы

Новая парадигма машинного обучения открывает новые возможности для решения задач управления нелинейными динамическими объектами. Введены необходимые теоретические основания машинного обучения, представлено формальное определение машинного обучения управления как поиска неизвестной функции управления. Машинное обучение позволяет создавать «интеллектуальные» контроллеры, способные использовать нелинейности системы для повышения производительности и хорошо работать в более широком диапазоне неопределенностей, чем классические подходы.

Машинное обучение управления позволяет находить хорошие решения, близкие к оптимальным, за ограниченное время. Однако в связи с новизной этих методов возникает необходимость в обосновании результатов, полученных с помощью машинного обучения. В этой главе вводятся некоторые теоретические основы машинного обучения управлению, представлены определения некоторых машинных свойств системы: вводится определение машинного обучения управлению, приводится машинное обоснование существования определенного свойства в некоторой математической модели.

Среди методов машинного обучения, с точки зрения задач синтеза управления, наилучшие результаты показывают методы символьной регрессии, в виду того, что они работают на алфавите элементарных функций и могут искать не только гладкие функции, как методы нейросетей, но и разрывные, в зависимости от заданного алфавита, что для управления весьма актуально. Разработанные методы символьной регрессии реализуют поиск математических выражений функции управления в виде определенного кода, в зависимости от метода, с помощью специального генетического алгоритма. В настоящей главе представлен общий подход методов символьной регрессии, а также специальный генетический алгоритм многокритериальной оптимизации на нечисловом пространстве кодов, который позволяет организовывать структурно-параметрический поиск функции управления.

Для организации вычислений на пространстве элементарных функций и избегания вычислительных ошибок, таких как переполнение разрядной сетки, в работе вводится новое пространство машинно реализуемых функций.

Методы символьной регрессии позволяют автоматизировать процесс синтеза систем управления, но сталкиваются с рядом трудностей, связанных со сложностью нечислового пространства поиска и отсутствие единой метрики в нем. В данном разделе представлены несколько разработанных методов символьной регрессии, модифицированных с помощью принципа малых вариаций базисного решения и успешно примененных для решения задач синтеза управления. Представлены предложенные уникальные типы малых вариаций и способы их кодирования. В работе разработаны программные комплексы реализации предложенных методов и представлены результаты проводимых экспериментов для синтеза системы стабилизации робототехническими объектами разработанными вариационными методами машинного обучения на основе символьной регрессии.

Глава 4. Применение принципа синтезированного оптимального управления в задачах управления робототехническими системами

В данной главе рассмотрим примеры решения различных задач оптимального управления робототехническими объектами на основе принципа синтезированного оптимального управления.

4.1. Колесный мобильный робот с дифференциальным приводом в условиях фазовых ограничений

В качестве первого робототехнического объекта управления рассмотрим колесный мобильный робот с дифференциальным приводом, представленный на рис. 4.1.

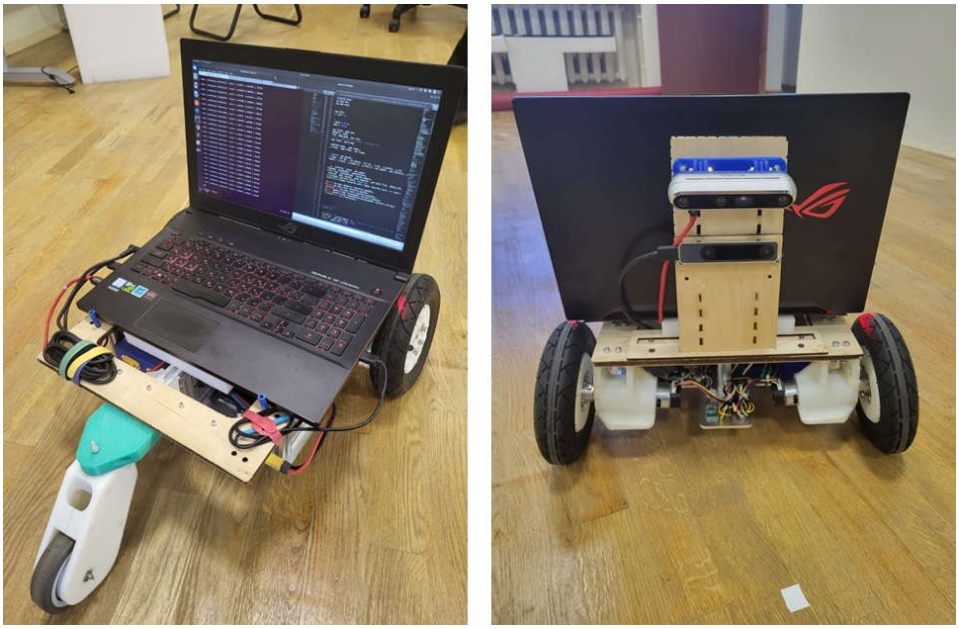

Рис.4.1. Колесный мобильный робот с дифференциальным приводом Роботоцентра ФИЦ ИУ РАН

Используется дифференциальная схема управления: робот движется вперед и назад, подавая одинаковое напряжение на правый и левый электродвигатели; поворот робота вправо или влево осуществляется за счет подачи большего напряжения на электродвигатель левого или правого колеса соответственно.

Рассмотрим задачу оптимального управления движением мобильного робота из некоторого начального состояния в конечное целевое положение в сложной среде с фазовыми ограничениями.

Сначала рассчитаем управление на модели.

В качестве математической модели используем следующую систему дифференциальных уравнений [226]:

$$\begin{aligned} \dot{x}_1 &= 0.5(u_1 + u_2)\cos(x_3), \\ \dot{x}_2 &= 0.5(u_1 + u_2)\sin(x_3), \\ \dot{x}_3 &= 0.5(u_1 - u_2), \end{aligned} \tag{4.1}$$

где управление имеет ограничения $-10 = u_i^- \leq u_i \leq u_i^+ = 10,\ i = 1,2.$

Начальное положение задано:

$$\mathbf{x}(0) = \mathbf{x}^0 = [0\ \ 0\ \ 0]^T. \tag{4.2}$$

Терминальное состояние задано:

$$\mathbf{x}(t_f) = \mathbf{x}^f = [10\ \ 10\ \ 0]^T, \tag{4.3}$$

где терминальное время t_f не задано, но ограничено, $t_f \leq t^+ = 2.$

Фазовые ограничения описываются следующими неравенствами:

$$\phi_i(\mathbf{x}) = r_i - \sqrt{\left(x_1 - x_{\{1,i\}}\right)^2 + \left(x_2 - x_{\{2,i\}}\right)^2} \leq 0, \tag{4.4}$$

где $i = 1,2$, $r_1 = r_2 = 2.5$, $x_{\{1,1\}} = 2.5$, $x_{\{2,1\}} = 2.5$, $x_{\{1,2\}} = 7.5$, $x_{\{2,2\}} = 7.5.$

Необходимо найти оптимальное управление с учетом ограничений на значения управления, такое, чтобы объект управления из заданного начального состояния достиг заданного конечного состояния, не наткнувшись на заданные препятствия за минимальное время.

Целевой функционал по быстродействию включает также штрафы за нарушение фазовых ограничений и за точность достижения терминальных условий.

$$J_1 = t_f + p_1 \sum_{\{i=1\}}^{2} \int_0^{t_f} \vartheta\left(\phi_i(\mathbf{x})\right) dt + p_2 \left\| \mathbf{x}(t_f) - \mathbf{x}^f \right\| \to \min, \tag{4.5}$$

где $p_1 = 3$, $p_2 = 1$, $\vartheta(a)$ – ступенчатая функция Хевисайда.

Сначала рассмотрим решение задачи оптимального управления как поиск функции управления в виде функции времени. Для решения традиционной задачи оптимального управления используется прямой подход. Для этого ось времени разбивается на интервалы $\Delta\, t = 0.2$.

$$K = \left\lfloor \frac{t^+}{\Delta \mathrm{t}} \right\rfloor = \left\lfloor \frac{2}{0.2} \right\rfloor = 10.$$

Функция управления для каждого интервала управления искалась в виде кусочно-линейной аппроксимации

$$u_i = \begin{cases} u_i^+ = 10, \text{если } \tilde{u}_i \geq u_i^+ \\ u_i^- = -10, \text{если } \tilde{u}_i \leq u_i^-, \\ \tilde{u}_i, \text{ иначе} \end{cases} \quad i = 1,2, \tag{4.6}$$

где

$$\tilde{u}_j = (q_{\{i+(j-1)K\}} - q_{\{i-1+(j-1)K\}}) \frac{t-(i-1)\Delta\, t}{\Delta\, t} + q_{\{i-1+(j-1)K\}}, \tag{4.7}$$

где $j = 1,2$, $(i-1)\Delta\, t \leq t < i\Delta\, t$, $i = 1, \dots, K$, $\mathbf{q} = \left[q_1 \dots\ q_{\{2K\}}\right]^T$ – вектор искомых параметров.

Всего нужно было найти $2K = 20$ параметров. Для нахождения оптимального вектора параметров можно использовать любой алгоритм нелинейного программирования. Поскольку за счет наличия фазовых ограничений функционал качества может быть неунимодальным и невыпуклым, применение градиентных методов ограничивается. Современные популяционные алгоритмы способны широко исследовать пространство поиска и находит глобальный минимум с большой вероятностью. В этой задаче был применен алгоритм оптимизации роя частиц (PSO).

В процессе поиска параметры имели ограничения $-20 \leq q_i \leq 20$, $i = 1, \dots, 20$.

В результате был найден следующий оптимальный вектор параметров:

$$\begin{gathered} \mathbf{q} = 18.8104 \quad 11.7582 \quad 19.2111 \quad 14.1466 \quad 14.8751 \\ 15.7302 \quad 18.8598 \quad 19.6042 \quad 19.9970 \quad -5.3989 \\ 17.9978 \quad 12.3394 \quad -1.3814 \quad 19.9329 \quad 18.0717 \\ 18.7211 \quad 6.9416 \quad 3.8498 \quad 5.9905 \quad 19.6724]^T . \end{gathered} \tag{4.8}$$

Значение функционала (4.5) составило $J_1 = 2.0044$. Траектория на плоскости $\{x_1, x_2\}$ представлена на рис. 4.2. Как видно, мобильный робот достиг конечного положения, не нарушая фазовых ограничений.

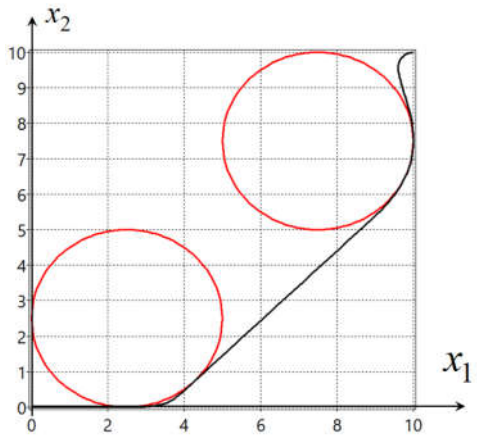

Рис.4.2. Траектория движения мобильного робота на плоскости $\{x_1, x_2\}$ при обычном оптимальном управлении.

Теперь решим ту же задачу оптимального управления на основе принципа синтезированного оптимального управления.

На первом этапе решается задача синтеза системы стабилизации. Система стабилизации синтезируется заранее и затем размещается на бортовом компьютере. Для синтеза был применен метод символьной регрессии сетевого оператора.

Вычислительное преимущество данного подхода состоит в том, что он использует принцип вариации базисного решения, что значительно ускоряет процесс поиска близкого к оптимальному решения. Метод кодирует возможные решения в виде квадратной верхней треугольной матрицы. Именно в виде матрицы, описывающей последовательность расчета функции управления, полученная система стабилизации помещается на бортовой компьютер.

При расчетах были заданы следующие параметры: ограничения на управление $-10 \leq u_i \leq 10$, $i = 1,2$, целевая точка стабилизации $\mathbf{x}^* = [x_1^* \ x_2^* \ x_3^*]^T = [0 \ 0 \ 0]^T$, область начальных положений определена сеткой из 26 положений

$$\overline{\mathrm{X}}_0 = \{\left[-2.5 \;\; -2.5 \;\; -\frac{5\pi}{12}\right]^T, [-2.5 \;\; -2.5 \;\; 0]^T,$$

$$\left[-2.5 \;\; -2.5 \;\; \frac{5\pi}{12}\right]^T, \left[-2.5 \;\; 0 \;\; -\frac{5\pi}{12}\right]^T, [-2.5 \;\; 0 \;\; 0]^T,$$

$$\left[-2.5 \;\; 0 \;\; \frac{5\pi}{12}\right]^T, \left[-2.5 \;\; 2.5 \;\; -\frac{5\pi}{12}\right]^T, [-2.5 \;\; 2.5 \;\; 0]^T,$$

$$\left[-2.5 \;\; 2.5 \;\; \frac{5\pi}{12}\right]^T, \left[0 \;\; -2.5 \;\; -\frac{5\pi}{12}\right]^T, [0 \;\; -2.5 \;\; 0]^T,$$

$$\left[0 \;\; -2.5 \;\; \frac{5\pi}{12}\right]^T, \left[0 \;\; 0 \;\; -\frac{5\pi}{12}\right]^T, \left[0 \;\; 0 \;\; \frac{5\pi}{12}\right]^T,$$

$$\left[0 \;\; 2.5 \;\; -\frac{5\pi}{12}\right]^T, [0 \;\; 2.5 \;\; 0]^T, \left[0 \;\; 2.5 \;\; \frac{5\pi}{12}\right]^T,$$

$$\left[2.5 \;\; -2.5 \;\; -\frac{5\pi}{12}\right]^T, [2.5 \;\; -2.5 \;\; 0]^T, \left[2.5 \;\; -2.5 \;\; \frac{5\pi}{12}\right]^T,$$

$$\left[2.5 \;\; 0 \;\; -\frac{5\pi}{12}\right]^T, [2.5 \;\; 0 \;\; 0]^T, \left[2.5 \;\; 0 \;\; \frac{5\pi}{12}\right]^T,$$

$$\left[2.5 \;\; 2.5 \;\; -\frac{5\pi}{12}\right]^T, [2.5 \;\; 2.5 \;\; 0]^T, \left[2.5 \;\; 2.5 \;\; \frac{5\pi}{12}\right]^T\}. \qquad (4.9)$$

Необходимо найти такую функцию управления

$$u_i = h_i(x_1^* - x_1, x_2^* - x_2, x_3^* - x_3, r_1, r_2, r_3),$$

где r_1, r_2, r_3 – постоянные параметры, $i = 1{,}2$, чтобы робот из всех 26 начальных условий (4.9) достигал точки стабилизации $\mathbf{x}^* = [x_1^* \;\; x_2^* \;\; x_3^*]^T = [0 \;\; 0 \;\; 0]^T$ с минимальным временем и максимальной точностью.

Вычисления проводились в режиме офлайн и затем помещались на борт. В результате синтеза методом сетевого оператора найдено решение, представленное в виде матрицы сетевого оператора и параметров.

Матрица сетевого оператора имела размерность 24×24, использовались двадцать элементарных функций с одним аргументом и две функции с двумя аргументами.

Результирующая функция управления (4.6) имела следующий вид:

$$\tilde{u}_1 = A^{-1} + \sqrt[3]{C} + \ exp(-|q_3\Delta_3|) + sgn(\Delta_3) + \mu(B), \qquad (4.10)$$

$$\tilde{u}_2 = \tilde{u}_1 + sin(\tilde{u}_1) + arctan(E) + \mu(B) + H - H^3 + sgn(\Delta_1),$$

$$A = C + B - B^3,$$

$$B = H + tanh(W) + \Delta_1,$$

$$C = tanh(D) + G^3 + H + sin(q_3\Delta_3),$$

$$D = E + H - H^3 + sgn(q_1\Delta_1) + arctan(q_1) + \vartheta(\Delta_3),$$

$$E \ = \ F + \sqrt[3]{H + tanh(W) + \Delta_1 + \sqrt[3]{\Delta_1}} + sgn(B) + arctan(q_1\Delta_1),$$

$$F \ = \ H + tanh(W) + \Delta_1 + \sqrt[3]{\Delta_1} + \ sgn(H + tanh(W) + \Delta_1) + W +$$

$$sgn(\ \Delta_1)\sqrt{q_i|\Delta_1|},$$

$$G \ = \ H + tanh(W) + \Delta_1 + \sqrt[3]{\Delta_1},$$

$$H \ = \ W + sgn(sgn(\Delta_1)q_2\Delta_2)\ exp(-|q_2\Delta_2|) + sin(\Delta_1),$$

$$W = sgn(\Delta_1)q_2\Delta_2 + q_3\Delta_3 + tanh(q_1\Delta_1),$$

$$\Delta_i = x_i^* - x_i, \ \ i = 1,2,3,$$

$$\mu(\alpha) = \begin{cases} \alpha, \text{если } \ |\alpha| \leq \ 1 \\ \ \text{sgn}(\alpha), \text{иначе} \end{cases},$$

$$r_1 = 14.72876, r_2 = 2.02710, r_3 = 4.02222.$$

Найденное решение (4.10) представлено на борту робота в виде матрицы сетевого оператора C_{NOP}, в форме которой и было найдено в процессе синтеза. Представление (4.10) в виде математического выражения является лишь репрезентативной формой. Матрица C_{NOP} представлена на рис.4.3.

Матрица кодирует нелинейные функции двух компонент управления. Каждый ненулевой элемент представляет собой либо унарную, либо бинарную операцию. Для расчета конкретных значений сигналов управления на каждом шаге матрица декодируется онлайн на бортовом компьютере в режиме реального времени. Для расчета сигналов управления на бортовой вычислительной машине алгоритм декодирования матрицы сетевого оператора был реализован на языке C++ и размещен в открытом репозитарии на github для общедоступного пользования [203].

```
PsiBasc:array [0..23,0..23] of integer=
((0,0,0,0, 0,0,1,10,  0,0,12,1,  15,0,0,0,  0,0,0,0,  0,0,0,10),
 (0,0,0,0,  0,0,0,1,  0,0,0,0,   0,0,0,0,  0,0,0,0,   0,0,0,0),
 (0,0,0,0,  0,0,0,0,  1,0,0,0,   0,0,0,9,  0,0,0,0,  10,0,0,0),
 (0,0,0,0,  0,0,1,0,  0,0,0,0,  0,0,0,13,  0,0,0,0,   0,0,0,0),

 (0,0,0,0,  0,0,0,1,  0,0,0,0,   0,0,0,0,  0,0,0,0,   0,0,0,0),
 (0,0,0,0,  0,0,0,0,  1,0,0,0,   0,0,0,0,  0,0,0,0,   0,0,0,0),
 (0,0,0,0,  0,0,2,0,  0,8,0,0, 0,4,13,10,  0,0,0,0,   0,0,0,0),
 (0,0,0,0,  0,0,0,2, 0,1,19,0,   0,0,0,0,  0,0,0,0,   0,0,0,0),

 (0,0,0,0,  0,0,0,0,  2,1,0,0,   0,0,0,0, 12,0,0,0,  19,0,0,0),
 (0,0,0,0,  0,0,0,0,  0,1,1,8,   0,1,0,0,  0,0,0,0,   0,0,0,0),
 (0,0,0,0,  0,0,0,0,  0,0,1,1,  0,0,4,23,  1,0,0,0,  0,0,0,23),
 (0,0,0,0,  0,0,0,0,  0,0,0,1, 1,10,10,0, 0,0,0,23, 0,16,0,16),

 (0,0,0,0,  0,0,0,0,  0,0,0,0,  1,1,15,0, 14,0,0,0,   0,0,0,0),
 (0,0,0,0,  0,0,0,0,  0,0,0,0,   0,1,1,0,  0,0,0,0,   0,0,0,0),
 (0,0,0,0,  0,0,0,0,  0,0,0,0,   0,0,1,1,  0,0,0,0,  0,0,0,13),
 (0,0,0,0,  0,0,0,0,  0,0,0,0,   0,0,0,1,  8,0,0,0,   0,0,0,0),

 (0,0,0,0,  0,0,0,0,  0,0,0,0,   0,0,0,0,  1,1,0,0,   0,0,0,0),
 (0,0,0,0,  0,0,0,0,  0,0,0,0,   0,0,0,0,  0,2,1,0,  15,0,0,0),
 (0,0,0,0,  0,0,0,0,  0,0,0,0,   0,0,0,0,  0,0,1,1,   0,0,0,0),
 (0,0,0,0,  0,0,0,0,  0,0,0,0,   0,0,0,0,  0,0,0,1,   5,0,0,0),

 (0,0,0,0,  0,0,0,0,  0,0,0,0,   0,0,0,0,  0,0,0,0,   1,1,0,0),
 (0,0,0,0,  0,0,0,0,  0,0,0,0,   0,0,0,0,  0,0,0,0,  0,1,1,12),
 (0,0,0,0,  0,0,0,0,  0,0,0,0,   0,0,0,0,  0,0,0,0,   0,0,2,1),
 (0,0,0,0,  0,0,0,0,  0,0,0,0,   0,0,0,0,  0,0,0,0,   0,0,0,1));
```

Рис.4.3. Матрица сетевого оператора C_{NOP}, описывающая систему стабилизации мобильного робота.

Для переноса расчетных управлений на реальный объект была разработана прикладная программная реализация системы управления робота в виде ROS-модуля. Для ее создания используется самая популярная на сегодняшний день робототехническая операционная система ROS (Robot operation system). Она предоставляет возможность работать со всеми аспектами системы управления, включая аппаратную абстракцию, низкоуровневое управление, передачу сообщений между процессами, и управление пакетами.

Разрабатываемые программные комплексы отрабатывались в имитационной среде-симуляторе Gazebo, которая является одним из

популярнейших робототехнических симуляторов, благодаря своей совместимости с ROS. Gazebo - это 3D-симулятор, цель которого смоделировать робота так, чтобы дать вам близкую замену тому, как робот будет вести себя в реальной физической среде. Gazebo имеет достаточно достоверную симуляцию физики и различных физических явлений, учитывает влияние сил, а также имеет большое количество плагинов для симуляции работы датчиков, например лидаров или камер. Благодаря этим фактам, большинство разработчиков систем управления робототехнических систем по всему миру, используют данную связку ROS/Gazebo как стандарт для отработки разрабатываемых алгоритмов управления [227, 228].

В настоящей работе была использована готовая Gazebo модель ROSbot 2.0, интегрированная в ROS [229]. В качестве источника позиции, использовался плагин, дающий истинные координаты робота. Модель робота в симуляторе представлена на рис. 4.4.

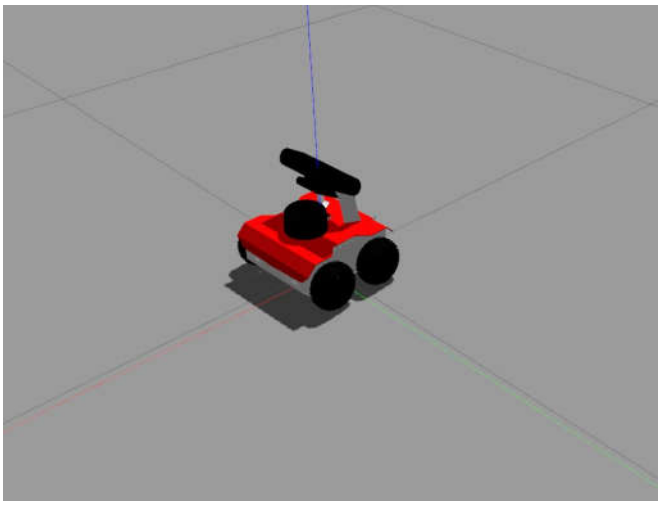

Рис.4.4. ROSbot 2.0 в симуляторе Gazebo

Синтезированный контроллер (4.6), (4.10) в виде матрицы сетевого оператора, представленной на рис. 4.3 проверен на работоспособность в симуляционной среде Gazebo в задаче стабилизации виртуальным роботом ROSbot. Была продемонстрирована стабильная и предсказуемая работа контроллера. Для реализации данного эксперимента система управления была

выполнена программно в виде ROS-модуля ROSbot Controller. Модуль реализует управление по принципу обратной связи на основе расчетной матрицы сетевого оператора. Узел контроллера ROSbot принимает на вход наземные координаты и цель движения робота, а на выходе формирует управляющие сигналы.

Разработанный модуль в дальнейшем может быть использован для различных робототехнических объектов управления для стабилизации относительно определенной точки пространства состояния по расчетной матрице сетевого оператора.

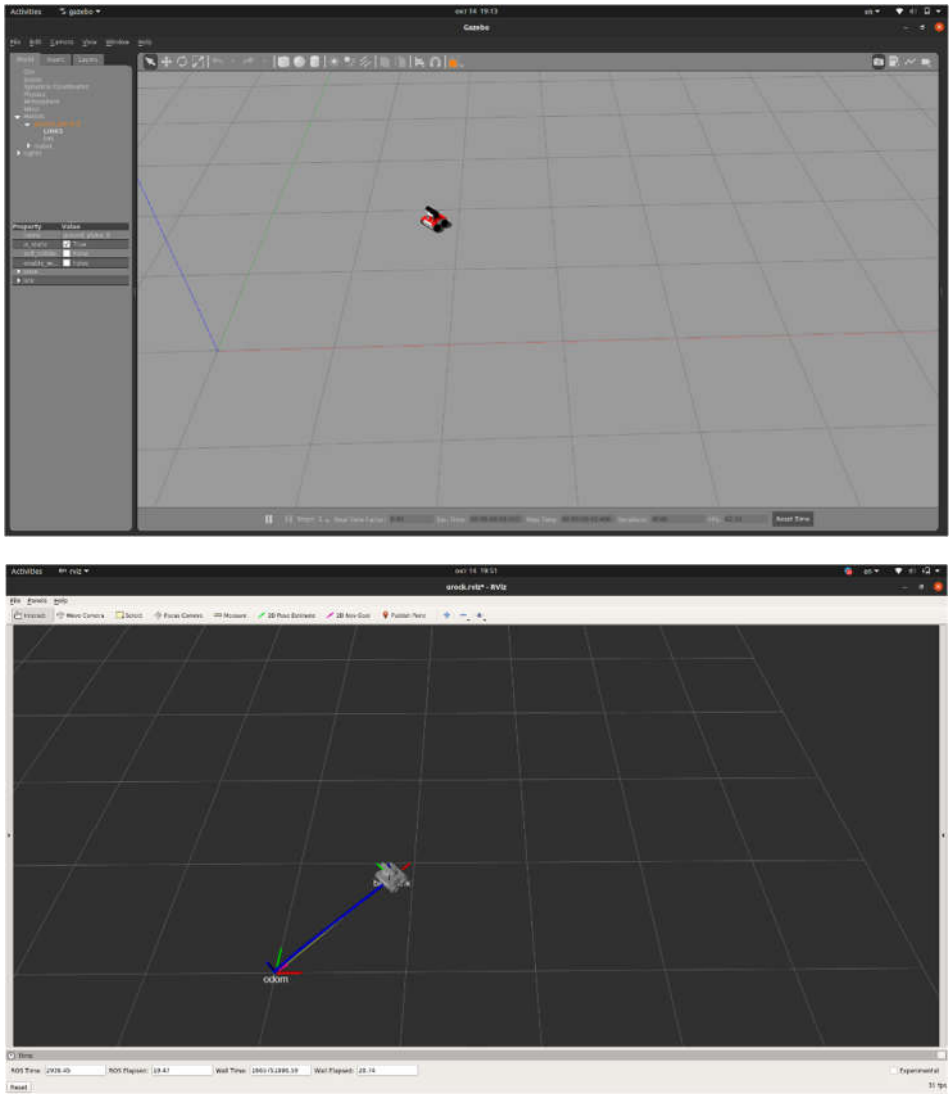

Рис.4.4. Тестирование ROS-модуля пространственной стабилизации в симуляторе

На рис.4.4 представлен пример испытаний робота в виртуальной среде с разработанным контроллером. Робот стабилизируется в заданную терминальную точку из различных задаваемых положений, не входивших в обучающее множество начальных условий, что является подтверждением эффективной работы системы стабилизации.

На втором этапе, согласно принципу синтезированного оптимального управления, ищется функция управления $\mathbf{x}^*(t)$. Эта функция управления ищется в виде кусочно-постоянной функции времени. Ось времени была разбита на $K = 20$ интервалов и по критерию (4.5) был найден оптимальный набор векторов $\mathbf{x}^* = [x_1^* \quad x_2^* \quad x_3^*]^T$ для каждого интервала времени. Нужно было найти 30 параметров.

В качестве алгоритма оптимизации использовался алгоритм роя частиц.

В процессе поиска применялись следующие ограничения на параметры:

$$-12 \leq x_1^* \leq 12, -12 \leq x_2^* \leq 12, -1.57 \leq x_3^* \leq 1.57.$$

В результате было найдено следующее решение

$$\mathbf{x}^{\{*,1\}} = [5.0847 \quad 1.220 \quad 1.57]^T, \tag{4.11}$$

$$\mathbf{x}^{\{*,2\}} = [11.5002 \quad 5.6125 \quad -1.5700]^T,$$

$$\mathbf{x}^{\{*,3\}} = [12.0 \quad 3.0270 \quad -0.3734]^T,$$

$$\mathbf{x}^{\{*,4\}} = [6.9277 \quad 2.5522 \quad 1.57]^T,$$

$$\mathbf{x}^{\{*,5\}} = [5.463 \quad 6.4091 \quad 0.8701]^T,$$

$$\mathbf{x}^{\{*,6\}} = [11.9407 \quad 7.1619 \quad 0.4311]^T,$$

$$\mathbf{x}^{\{*,7\}} = [8.1942 \quad 7.4549 \quad 1.5687]^T,$$

$$\mathbf{x}^{\{*,8\}} = [12.0 \quad 4.6526 \quad 1.570]^T,$$

$$\mathbf{x}^{\{*,9\}} = [12.0 \quad 7.5388 \quad 1.4516]^T,$$

$$\mathbf{x}^{\{*,10\}} = [12.0 \quad 6.9478 \quad 1.570]^T.$$

Значение функционала (4.5) для найденного решения составило $J_1 = 2.00$. Траектория на плоскости $\{x_1, x_2\}$ представлена на рис. 4.5. Как видно из рисунка, мобильный робот достиг конечного положения без нарушения заданных фазовых ограничений. На рис. 4.5 маленькие черные квадратики – это проекции найденных векторов $\mathbf{x}^{\{*,i\}}$, $i = 1, \ldots, 10$ на плоскость $\{x_1, x_2\}$.

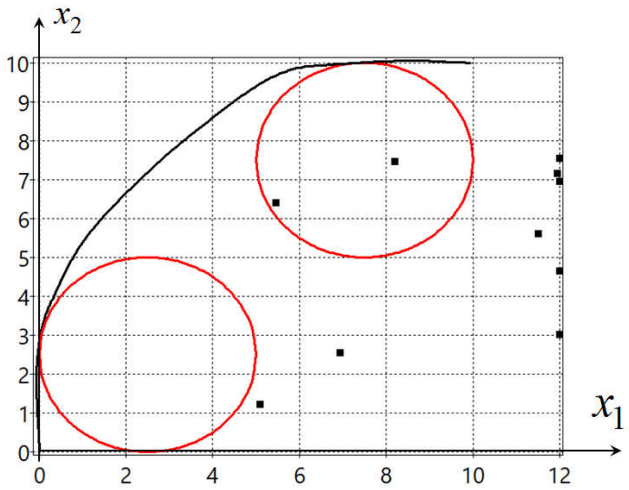

Рис.4.5. Траектория движения мобильного робота на плоскости $\{x_1, x_2\}$ *с синтезированным оптимальным управлением.*

Для оценки реализуемости полученных управлений введем возмущения в модель

$$\begin{aligned} \dot{x}_1 &= 0.5(u_1 + u_2)\cos(x_3) + b\xi(t), \\ \dot{x}_2 &= 0.5(u_1 + u_2)\sin(x_3) + b\xi(t), \\ \dot{x}_3 &= 0.5(u_1 - u_2) + b\xi(t), \end{aligned} \tag{4.12}$$

и в начальные условия

$$x_i(0) = x_i^0 + b_0\xi(t),\ i = 1,2,3. \tag{4.13}$$

Результаты экспериментов представлены в таблице 4.1.

Таблица 4.1. Проверка влияния неопределенностей.

b	b_0	J_{dir}	$\sigma(J_{dir})$	J_{synt}	$\sigma(J_{synt})$
0.05	0	2.2625	0.4417	2.0977	0.0630
0	0.05	2.9076	1.1029	2.4359	0.2942
0.05	0.05	3.9642	1.3065	2.312	0.2880
0.1	0	2.1824	0.2824	2.1218	0.0698
0	0.1	3.459	1.0014	2.4914	0.2993
0.1	0.1	4.4154	1.5939	2.4932	0.4659

Было проведено по десять испытаний для каждого значения b и b_0. В таблице J_{dir} – среднее значение функционала (4.5), а $\sigma(J_{\text{dir}})$ – его среднеквадратичное отклонение для возмущенной модели (4.12) и возмущенного начального условия (4.13) с оптимальным управлением (4.6)-(4.8), полученным прямым методом с помощью кусочно-линейной аппроксимации функции управления, J_{synt} и $\sigma(J_{\text{synt}})$ – соответственно среднее значение функционала (4.5) и его среднеквадратичное отклонение при синтезированном управлении (4.10) и (4.11).

Как видно из представленных в таблице результатов традиционное оптимальное управление очень чувствительно к возмущениям, особенно в начальных условиях. Это означает, что в таком виде оно не подходит для реализации на реальных объектах. В этом проявляется основная суть предложенного принципа, заключающегося в поиске оптимального управления в классе управлений, реализуемых на борту и нечувствительных к некоторым возмущениям за счет включения системы стабилизации в модель объекта управления.

С данным объектом проводилось много различных экспериментальных исследований с различными параметрами алгоритмов, моделей, фазовых ограничений, результаты которых представлены в публикация автора.

В следующей части отдельно рассмотрен случай, когда объектом управления является группа подобных мобильных роботов.

4.2. Задача управления группой мобильных роботов

Рассмотрим задачу оптимального управления группой из N роботов с фазовыми ограничениями. Для задачи характерно наличие двух видов ограничений: статических и динамических, что значительно усложняет постановку задачи. В данном разделе представлено экспериментальное сравнение метода синтезированного управления с наиболее популярным прямым численным методом сведения задачи оптимального управления к

нелинейному программированию. Проведено исследование полученных управлений в присутствии шумов.

Математическая модель объекта управления описывается следующей системой уравнений, включающей случайную компоненту, имитирующую шумы или неточности модели:

$$\begin{aligned} \dot{x}_j &= 0.5(u_{1,j} + u_{2,j})\cos(\theta_j) + \beta\xi(t), \\ \dot{y}_j &= 0.5(u_{1,j} + u_{2,j})\sin(\theta_j) + \beta\xi(t), \\ \dot{\theta}_j &= 0.5(u_{1,j} - u_{2,j}) + \beta\xi(t), \end{aligned} \tag{4.14}$$

где $j = 1,...,N$, x_j, y_j, θ_j - компоненты вектора состояния группы роботов, $\mathbf{u}_j = [u_{1,j}, u_{2,j}]^T$ - вектор управления группы роботов, β - постоянный положительный параметр, $\xi(t)$ - случайная функция, которая принимает значения от -1 до 1.

Заданы ограничения на компоненты вектора управления

$$u_i^- \le u_i^j \le u_i^+, \; i = \overline{1,m}, \; j = \overline{1,N}. \tag{4.15}$$

где u_i^-, u_i^+ - заданные значения.

Задано начальное положение каждого робота, включающее случайную компоненту с коэффициентом γ

$$\begin{aligned} x_j(0) &= x_j^0 + \gamma\xi(0), \\ y_j(0) &= y_j^0 + \gamma\xi(0), \\ \theta_j(0) &= \theta_j^0 + \gamma\xi(0), \end{aligned} \tag{4.16}$$

где γ - заданная положительная константа.

Задано целевое терминальное состояние

$$\begin{aligned} x_j(t_f) &= x_j^f, \\ y_j(t_f) &= y_j^f, \\ \theta_j(t_f) &= \theta_j^f. \end{aligned} \tag{4.17}$$

Заданы статические фазовые ограничения

$$\varphi_i(x_j, y_j) = r_i^2 - (x_i^* - x_j)^2 - (y_i^* - y_j)^2 \le 0,\ j = 1,...,N, \qquad (4.18)$$

где r_i, x_i^*, y_i^* - заданные параметры статических фазовых ограничений, $i = 1,...,S$, S - количество фазовых ограничений.

Учтем возможные столкновения роботов между собой в виде динамических фазовых ограничений:

$$\psi_{i,k} = d^2 - (x_i - x_k)^2 - (y_i - y_k)^2 \le 0, \qquad (4.19)$$

где $i = 1,...N-1$, $k = i+1,...N$, d - заданная положительная величина, определяющая максимальный габаритный размер робота.

Задан критерий качества управления, минимизирующий время достижения цели и включающий штрафные функции за отклонение от терминального состояния и за нарушение фазовых ограничений:

$$J = t_f + \alpha_1\left(\sum_{j=1}^{N}(x_j(t_f) - x_j^f) + \sum_{j=1}^{N}(y_j(t_f) - y_j^f) + \sum_{j=1}^{N}(\theta_j(t_f) - \theta_j^f)\right) +$$

$$+\alpha_2\left(\int_0^{t_f}\sum_{j=1}^{N}\sum_{i=1}^{S}\vartheta(\varphi_i(x_j, y_j))dt\right) + \alpha_3\left(\int_0^{t_f}\sum_{i=1}^{N-1}\sum_{j=i+1}^{N}\vartheta(\psi_{i,k})dt\right) \to \min \qquad (4.20)$$

где t_f - время процесса управления

$$t_f = \begin{cases} t, & \text{если } t < t^+ \text{ и } \sum_{j=1}^{N}\delta_j(t) < \varepsilon, \\ t^+ & \text{- иначе} \end{cases},$$

$$\delta_j(t) = \sqrt{(x_j - x_j^f)^2 + (y_j - y_j^f)^2 + (\theta_j - \theta_j^f)^2},$$

ε - малая положительная величина, t^+ - максимально возможное время управления, $\alpha_1, \alpha_2, \alpha_3$ - весовые коэффициенты штрафных функций, $\vartheta(a)$ - функция Хэвисайда

$$\vartheta(a) = \begin{cases} 1, & \text{если } a > 0 \\ 0 & \text{- иначе} \end{cases}.$$

Поставленную задачу управления (4.14) – (4.20) решаем с помощью двух подходов (прямого и синтезированного) и исследуем полученные решения на чувствительность к неточностям модели и начальных условий.

Первоначально решим поставленную задачу (4.14) – (4.20) на основе принципа синтезированного оптимального управления.

Согласно принципу, сначала стабилизируем объект относительно некоторой точки пространства состояний. Поскольку в рассматриваемой постановке роботы однотипны, решим задачу синтеза системы стабилизации для одного робота и без учета фазовых ограничений (4.18).

Подробно этап синтеза системы стабилизации для данного типа мобильного робота был рассмотрен в предыдущем разделе и в результате получена система управления по состоянию (4.10) и соответствующий бортовой контроллер.

Полученные функции управления (4.10), обеспечивающие стабилизацию объекта, подставляем в уравнения модели (4.14) и далее для получения оптимальных траекторий движения роботов находим K векторов $(\overline{x}^1,\overline{y}^1,\overline{\theta}^1),...,(\overline{x}^K,\overline{y}^K,\overline{\theta}^K)$ точек стабилизации. Для поиска оптимального расположения точек стабилизации используем эволюционный алгоритм роя частиц.

В рассматриваемой задаче использовали следующие значения параметров: количество роботов $N=4$, величина интервала $\Delta t=0.7$, при максимально допустимом времени управления $t^+=2.5$, количество точек стабилизации для каждого робота $K=3$, количество фазовых ограничений $S=4$, координаты центра и габаритные параметры ограничений $r_i=2.5$, $i=1,2,3,4$, $x_1^*=2$, $y_1^*=5$, $x_2^*=5$, $y_2^*=8$, $x_3^*=5$, $y_3^*=8$, $x_4^*=5$, $y_4^*=2$, $d=2$. Значения искомых компонент векторов точек стабилизации имели следующие ограничения: $\overline{x}_i^+=12$, $\overline{y}_i^+=12$, $\overline{\theta}_i^+=\pi/2$, $\overline{x}_i^-=-1$, $\overline{y}_i^-=-1$, $\overline{\theta}_i^-=-\pi/2$.

Начальные условия для роботов имели следующие значения: $x_0^1 = 0$, $y_0^1 = 0$ $\theta_0^1 = 0$, $x_0^2 = 0$, $y_0^2 = 10$, $\theta_0^2 = 0$, $x_0^3 = 10$, $y_0^3 = 0$, $\theta_0^3 = 0$, $x_0^4 = 10$, $y_0^4 = 10$, $\theta_0^2 = 0$, терминальные условия: $x_f^1 = 10$, $y_f^1 = 10$, $\theta_f^1 = 0$, $x_f^2 = 10$, $y_f^2 = 0$, $\theta_f^2 = 0$, $x_f^3 = 0$, $y_f^3 = 10$, $\theta_f^3 = 0$, $x_f^4 = 0$, $y_f^4 = 0$, $\theta_f^2 = 0$.

В результате было получено следующее оптимальное решение:

$$\bar{\mathbf{x}}^1 = [-0.989, 10.404, -0.535]^T,\ \bar{\mathbf{x}}^2 = [5.998, 5.535, -1.214]^T,$$

$$\bar{\mathbf{x}}^3 = [1.572, 0.381, -0.584]^T,\ \bar{\mathbf{x}}^4 = [9.711, 7.629, -0.069]^T,$$

$$\bar{\mathbf{x}}^5 = [7.003, 5.010, 0.782]^T,\ \bar{\mathbf{x}}^6 = [5.107, 2.345, 0.148]^T,$$

$$\bar{\mathbf{x}}^7 = [7.081, 6.193, -0.035]^T,\ \bar{\mathbf{x}}^8 = [4.613, 3.878, 0.082]^T,$$

$$\bar{\mathbf{x}}^9 = [7.009, 4.984, -0.765]^T,\ \bar{\mathbf{x}}^{10} = [9.442, 2.000, -0.184]^T,$$

$$\bar{\mathbf{x}}^{11} = [7.776, 2.214, -1.215]^T,\ \bar{\mathbf{x}}^{12} = [7.394, 2.422, -0.465]^T.$$

График траекторий движения всех четырех роботов на плоскости приведен на рис. 4.6. Значение функционала составило $J = 2.8590$.

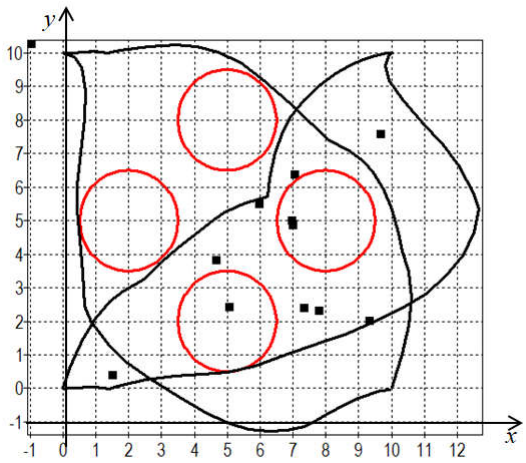

Рис.4.6. Траектория движения роботов, полученная на основе принципа синтезированного оптимального управления.

Для сравнения та же самая задача оптимального управления (4.14) – (4.20) была решена прямым методом. Управление аппроксимировалось кусочно-линейной функцией на интервалах.

$$u_i^j = \begin{cases} u_i^+, \text{ если } q(t,j,k,\Delta t) \ge u_i^+, \\ u_i^-, \text{ если } q(t,j,k,\Delta t) \le u_i^-, \\ q(t,j,k,\Delta t) \text{ иначе,} \end{cases} \tag{4.21}$$

где

$$q(t,j,k,\Delta t) = q_{(i-1)M+k}(i - t/\Delta t) + q_{(i-1)M+k}(t/\Delta t - k + 1),$$

$$k\Delta t \le t < (k+1)\Delta t,\ k = 1,...,M,\ j = 1,...,8,\ M = \left\lfloor t^+ / \Delta t \right\rfloor.$$

Вектор параметров q имел размерность 88, значение интервала составляло 0,25 сек, предельное время процесса управления составляло 2,5 сек. Ограничения по параметрам были $-20 \le q_i \le 20,\ i = 1,...,88$.

В результате оптимизации PSO-алгоритм нашел оптимальное решение со значением функционала $J = 3.6100$. Траектории движения роботов представлены на рис.4.7.

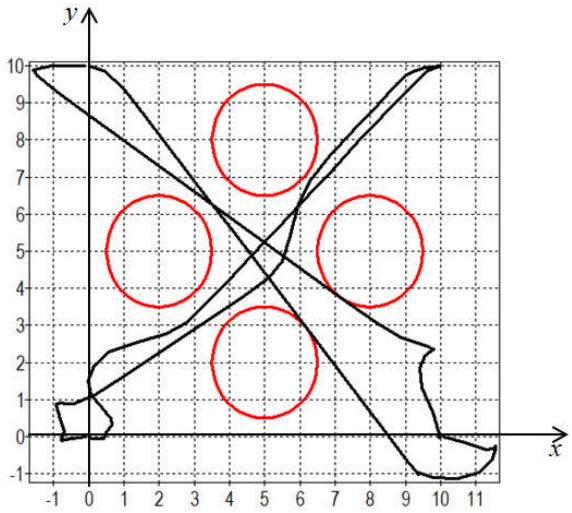

Рис.4.7. Оптимальная траектории движения роботов, полученная методом кусочно-линейной аппроксимации и алгоритмом PSO.

Теперь исследуем чувствительность полученных решений к возмущениям. Для этого мы промоделировали систему с полученными управлениями, увеличив коэффициенты шума β и γ.

Таблица 4.2: Значения функционала качества для полученных решений в присутствии возмущений в модели

β	**Синтезированное управление**	**Кусочно-линейная аппроксимация**
1	2.9840	4.8833
5	3.5094	6.7903
10	3.8064	10.0581

Таблица 4.3: Значения функционала качества для полученных решений в присутствии возмущений в начальных условиях

γ	**Синтезированное управление**	**Кусочно-линейная аппроксимация**
0.1	3.5309	5.2930
0.5	5.7881	9.9566
1	7.1876	11.4535

Для каждого уровня шума мы сделали 10 тестов, определяющих средние значения функционала (5.20). Таблица 4.2 содержит результаты для возмущений в модели, а таблица 4.3 для возмущений в начальных условиях.

На рисунках 4.8 – 4.11 представлены траектории движения роботов с тем же управлением, полученным прямым и синтезированным подходами, но при наличии шума.

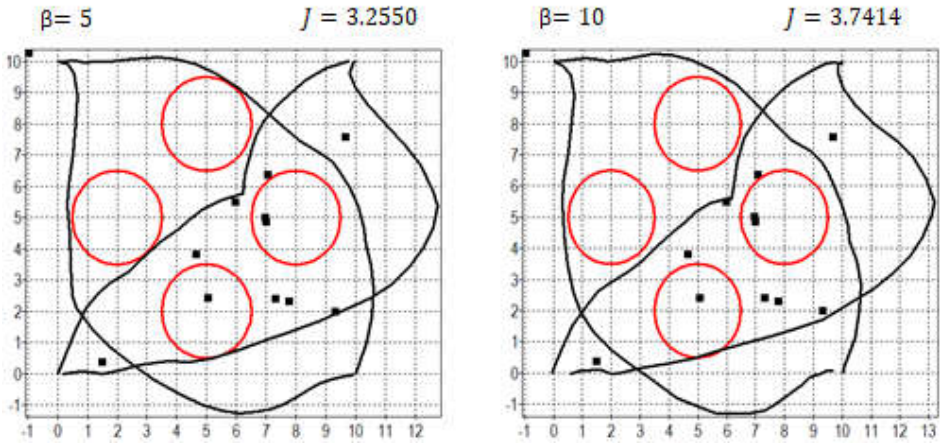

Рис.4.8. Траектории движения роботов, полученные методом синтезированного оптимального управления, при наличии возмущений в модели.

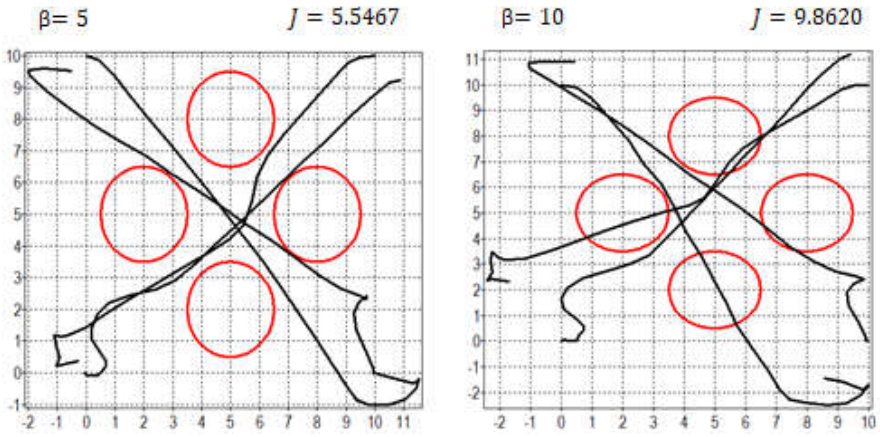

Рис.4.9. Траектории движения роботов, полученные методом кусочно-линейной аппроксимации и алгоритмом PSO, при наличии возмущений в модели.

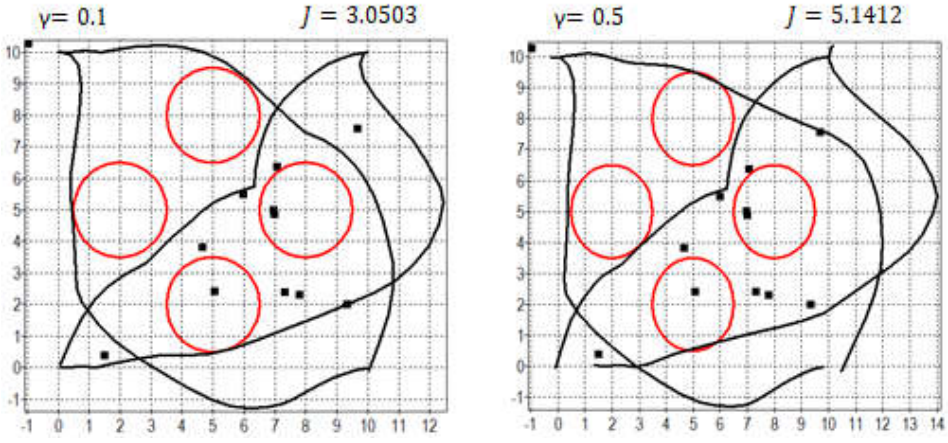

Рис.4.10. Траектории движения роботов, полученные методом синтезированного оптимального управления, при наличии возмущений в начальных условиях.

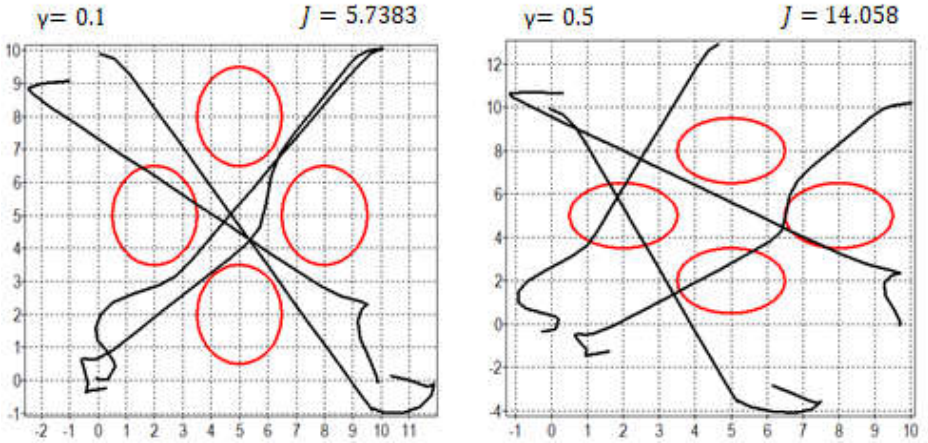

Рис.4.11. Траектории движения роботов, полученные методом кусочно-линейной аппроксимации и алгоритмом PSO, при наличии возмущений в начальных условиях.

Исследование полученных управлений на чувствительность к возмущениям показало меньшую чувствительность к возмущениям системы с синтезированным управлением, особенно при возмущениях в начальных условиях.

Другие проводимые эксперименты с однородными группами мобильных роботов представлены в работах автора [230, 231].

4.3. Задача управления всенаправленным мобильным роботом с колесами типа mecanum

Синтезированный подход к оптимальному управлению применяется в данном разделе к мобильному роботу на mecanum колесах (см. рис.4.12), конструкция которых позволяет роботу двигаться под любым углом к направлению его оси без поворота.

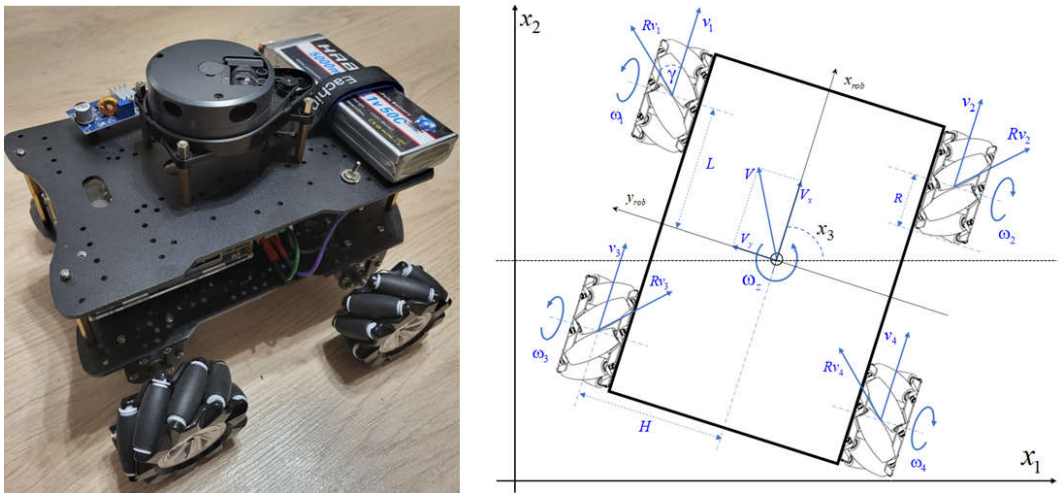

Рис.4.12. Мобильный робот на тесапит колесах

Рассмотрим задачу оптимального управления, в которой два одинаковых меканум-робота должны поменяться местами за минимальное время в некотором заданном пространстве с препятствиями, избежав столкновений как с препятствиями, так и друг с другом [232, 233].

Математическая модель объекта управления описывается следующей системой дифференциальных уравнений:

$$\dot{x_1} = 0.25((u_1 + u_4)(\cos(x_3) + \sin(x_3)) + (u_2 + u_3)(\cos(x_3) - \sin(x_3))), \quad (4.22)$$

$$\dot{x_2} = 0.25((u_1 + u_4)(\sin(x_3) - \cos(x_3)) + (u_2 + u_3)(\cos(x_3) + \sin(x_3))),$$

$$\dot{x_3} = \frac{0.25(-u_1 + u_2 - u_3 + u_4)}{L_0 + H_0},$$

$$\dot{x_4} = 0.25((u_5 + u_8)(\cos(x_6) + \sin(x_6)) + (u_6 + u_7)(\cos(x_6) - \sin(x_6))),$$

$$\dot{x_5} = 0.25((u_5 + u_8)(\sin(x_6) - \cos(x_6)) + (u_6 + u_7)(\cos(x_6) + \sin(x_6))),$$

$$\dot{x_6} = \frac{0.25(-u_5 + u_6 - u_7 + u_8)}{L_0 + H_0},$$

где x_1, x_2, x_3 – фазовые координаты первого робота, x_4, x_5, x_6 – фазовые координаты второго робота, u_1, u_2, u_3, u_4 – компоненты вектора управления для первого робота, u_5, u_6, u_7, u_8 – компоненты вектора управления для второго робота, L_0, H_0 – геометрические параметры роботов, $L_0 = 2$, $H_0 = 1$.

Особенностью рассматриваемой математической модели (4.22) является избыточное управление, $m = 8$ компонент вектора управления и $n = 6$ компонент вектора состояния, $m > n$.

Управление имеет ограничения

$$-10 = u_i^- \le u_i \le u_i^+ = 10, \quad i = 1, \dots, 8, \tag{4.23}$$

где u_i^- и u_i^+ – минимальное и максимальное значения управлений.

Зададим начальное положение объектов

$$\mathbf{x}(0) = \mathbf{x}^0 = [0 \;\; 0 \;\; 0 \;\;\; 10 \;\; 10 \;\;\; 0]^T, \tag{4.24}$$

и целевое терминальное положение

$$\mathbf{x}(t_f) = \mathbf{x}^f = [10 \;\; 10 \;\;\; 0 \;\; 0 \;\; 0 \;\; 0]^T, \tag{4.25}$$

где t_f – время достижения терминального состояния, которое не задано, но ограничено, и определяется из соотношения

$$t_f = \begin{cases} t, & \text{если } t < t^+ \text{ и } \|\mathbf{x}^f - \mathbf{x}(t_f)\| < \varepsilon_0, \\ t^+, & \text{иначе,} \end{cases}$$

$\varepsilon_0 = 0.05$, $t^+ = 1.9$.

Качество управление определяется по функционалу, который включает в себя время, штрафы за несоблюдение фазовых ограничений и точность достижения конечного состояния.

$$J_1 = t_f + p_1\|\mathbf{x}(t_f) - \mathbf{x}^f\| + \sum_{\{i=1\}}^{K} p_2 \int_0^{t_f} \vartheta(\phi_i(\mathbf{x}))dt + \\ + p_3 \int_0^{t_f} \vartheta(\chi(\mathbf{x}))dt \to \min, \tag{4.26}$$

где p_1, p_2, p_3 – весовые коэффициенты, $p_1 = 1$, $p_2 = 3$, $p_3 = 3$, $i = 1, \dots, K$, $K = 8$, $\vartheta(a)$ – функция Хевисайда,

$$\vartheta(a) = \begin{cases} 1, & \text{если a} > 0, \\ 0, & \text{иначе.} \end{cases}$$

Фазовые ограничения имели следующие параметры:

$$\phi_i(\mathbf{x}) = r_i - \sqrt{\{(x_1) - x_{1,i}\}^2 + \left(x_2 - x_{2,i}\right)^2},\ i = 1,2,3,4,$$

$$\phi_i(\mathbf{x}) = r_{i-4} - \sqrt{\{(x_4) - x_{1,i-4}\}^2 + \left(x_5 - x_{2,i-4}\right)^2},\ i = 5,6,7,8, \quad (4.27)$$

$$\chi(\mathbf{x}) = r_0 - \sqrt{\{(x_1 - x_4)^2 + (x_2 - x_5)^2\}}, \quad (4.28)$$

$r_1 = 2,\ r_2 = 2.5,\ r_3 = 2.5,\ r_4 = 2,\ x_{1,1} = 2,\ x_{2,1} = 2,\ x_{1,2} = 8,\ x_{2,2} = 2,\ x_{1,3} = 2,$ $x_{2,3} = 8,\ x_{1,4} = 8,\ x_{2,4} = 8,\ r_0 = 1.$

Согласно принципу синтезированного оптимального управления, первоначально решается задача синтеза системы стабилизации в обратной связи, чтобы замкнутая система управления была не просто устойчива относительно некоторой заданной точки равновесия в пространстве состояний, но и все решения из некоторой заданной области сходились в точку равновесия с минимальным значением критерия качества, в частности функционала по быстродействию. Для синтеза используется один из методов символьной регрессии, метод сетевого оператора, который позволяет осуществлять машинное обучение системы управления без учителя, только на основе минимизации значения функционала.

Для численного синтеза системы стабилизации необходимо было достичь заданной терминальной точки $\mathbf{x}^* = [0\ \ 0\ \ 0]^T$ из 18 начальных положений

$$\mathrm{X}_0 = \left[-5 + id_1 \quad -5 + jd_2 \quad -\frac{5\pi}{12} + kd_3\right]^T, \quad (4.29)$$

где $d_1 = 5,\ d_2 = 10,\ d_3 = \frac{5\pi}{12},\ i = 0,1,2,\ j = 0,1,\ k = 0,1,2.$

Это означает, что мы берем некую область начальных условий и находим такую функцию обратной связи, которая будет достигать заданного конечного положения из различных точек этой области с наименьшим значением критерия качества стабилизации:

$$J = \max\{t_{f,1}, \dots, t_{f,K}\} + a_1 \sum_{i=1}^{K} \left\|\mathbf{x}^f - \mathbf{x}\left(t_{f,i}, \mathbf{x}^{0,i}\right)\right\| \to \min, \quad (4.30)$$

где a_1 – весовой коэффициент, $t_{f,i}$ – время достижения заданного терминального положения из начального положения $\mathbf{x}^{0,i}$ из множества начальных положений (4.29) $\mathrm{X}_0 = \{\mathbf{x}^{0,1}, \dots, \mathbf{x}^{0,K}\}$, $i \in \{1, \dots, K\}$.

Так как роботы одинаковые, делаем синтез системы стабилизации для одного робота, и повторяем его для другого.

В результате была получена следующая функция управления

$$u_{i+(j-1)2} = \begin{cases} u^{+}_{i+(j-1)2}, & \text{если} \quad \tilde{u}_{i+(j-1)2} > u^{+}_{i+(j-1)2}, \\ u^{-}_{i+(j-1)2}, & \text{если} \quad \tilde{u}_{i+(j-1)2} < u^{-}_{i+(j-1)2}, \\ \tilde{u}_{\mathrm{i+(j-1)2}}, & \text{иначе,} \end{cases} \tag{4.31}$$

где $i = 1{,}2$, $j = 1{,}2$, а значения $\tilde{u}_{\mathrm{i+(j-1)2}}$ вычисляются по полученной матрице сетевого оператора (4.32). Матрица сетевого оператора $\boldsymbol{\Psi}$ позволяет вычислять значения управлений в зависимости от значений состояния объекта в каждый момент времени.

$$\boldsymbol{\Psi} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 10 & 0 & 4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 19 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 8 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 13 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 5 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 15 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 1 & 6 & 0 & 12 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 1 & 1 & 23 & 2 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 12 & 6 & 5 & 10 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 10 & 18 & 23 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 15 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 15 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \tag{4.32}$$

$$q_1 = 11.89282,\ q_2 = 10.15381,\ q_3 = 15.25903.$$

Графики траекторий движения робота из четырех начальных состояний в заданное терминальное положение представлены на рис. 4.13.

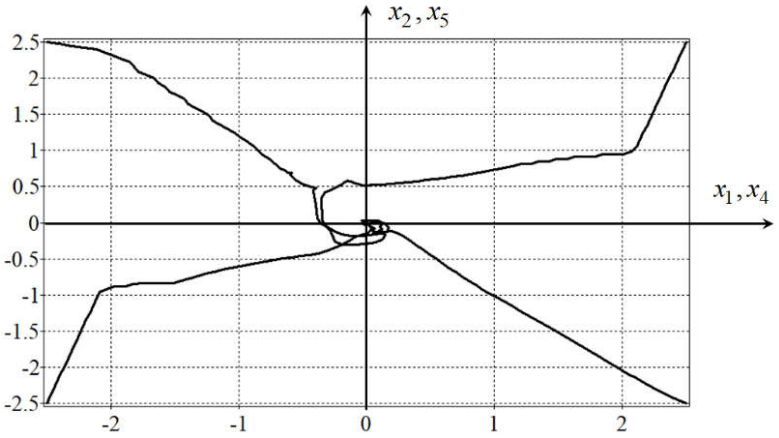

Рис.4.13. Траектории движения объекта из 4х начальных условий в терминальное положение с помощью системы стабилизации на основе сетевого оператора (4.32)

На втором этапе решается задача оптимального управления с функционалом (4.26) с учетом фазовых ограничений для замкнутой системы с функцией управления (4.31)-(4.32). Управлением на втором этапе является вектор $\mathbf{x}^*$, определяющий положение точки устойчивого равновесия в пространстве состояний в каждый момент времени. Для решения задачи оптимального управления ось времени разбивается на интервалы, и на каждом интервале функция управления аппроксимируется кусочно-постоянной функцией.

Значение интервала положим $\Delta\, t = 0.19$.

$$x_i^* = x_i(t_j) = q_{\{i+(j-1)6\}},$$

где $i = 1, \dots, 6$, $t_j = (j-1)\Delta\, t$, $j = 1, \dots, D$, D – количество интервалов, $D = t^+/\Delta t = 1.9/\,0.19 = 10$.

Итого необходимо было найти оптимальный вектор с $10 \cdot 6 = 60$ параметрами

$$\mathbf{q} = [q_1 \dots q_{60}]^T.$$

Для решения задачи был применен эволюционный алгоритм роя частиц.

Были найдены следующие значения искомых параметров управления:

$$\mathbf{q} = [-2.4862 \quad 2.0000 \quad 0.6231 \quad -2.1975 \quad -1.3799 \; 0.7058$$
$$-2.5000 \quad 1.9421 \quad 1.2094 \quad -0.6933 \quad -2.0088 \quad -0.3378$$
$$0.1956 \quad 1.7643 \quad 0.6378 \quad 1.4052 \quad -2.4611 \quad -0.3230$$
$$1.7245 \quad 2.0000 \quad 0.8872 \quad 1.6769 \quad -1.2832 \quad -0.4372$$
$$0.5624 \quad 1.9987 \quad -1.1601 \quad 1.9452 \quad -1.8859 \quad 0.7357$$
$$1.6876 \quad 1.5024 \quad 0.0997 \quad 1.1642 \quad -1.3678 \quad 0.5321$$
$$1.6945 \quad 1.9946 \quad 0.5580 \quad 0.7886 \quad -1.6027 \quad 1.3090$$
$$1.1795 \quad 1.5385 \quad -1.0798 \quad 0.6781 \quad -1.9975 \quad 0.0204$$
$$-0.8939 \quad 1.0578 \quad -0.4106 \quad -2.1003 \quad -1.2544 \quad -1.3090$$
$$-2.5000 \quad 1.0746 \quad -1.2216 \quad -2.0840 \quad -1.3344 \quad -0.8913]^T \qquad (4.33)$$

Оптимальное значение функционала (4.26) составило $J = 1{,}923$.

Оптимальные траектории на горизонтальной плоскости двух роботов представлены на рис. 4.14. Сплошная линия – траектория первого робота, штриховая линия – траектория второго робота, красные окружности – заданные фазовые ограничения, черные квадратики – проекции управления $\mathbf{x}^*$ на горизонтальную плоскость для первого робота, белые квадратики — проекции управления $\mathbf{x}^*$ на горизонтальную плоскость для второго робота.

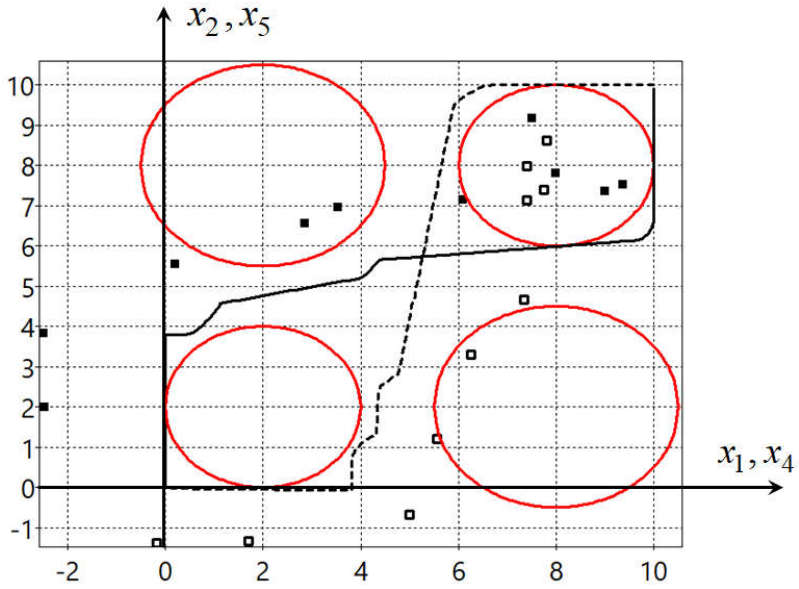

Рис.4.14. Траектории движения объекта на плоскости при синтезированном управлении

На рис. 4.14 очень хорошо видно, что точки равновесия расположены не на траектории движения робота, как это делается при обычной стабилизации, а вне траектории. Расставляя точки на траектории, мы можем потерять в качестве, так, например, при приближении к точке равновесия робот должен замедляться. При синтезированном подходе расположение точек обеспечивает минимизацию значения функционала.

На рисунках 4.15 - 4.20 показаны графики оптимальных значений компонент вектора пространства состояний (черные линии) и соответствующих значений вектора $\mathbf{x}^*$ (красные линии).

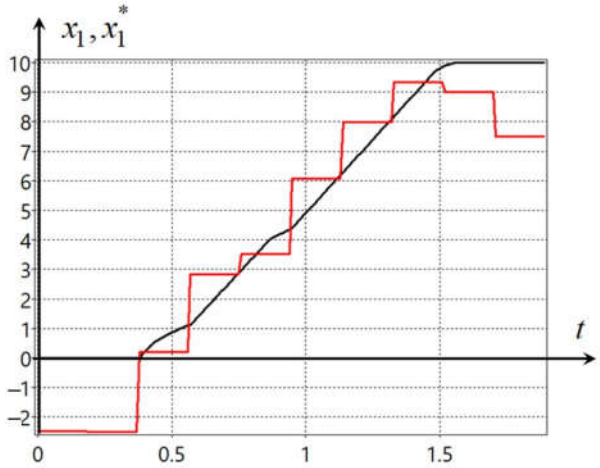

Рис. 4.15. График фазовой переменной x_1 и управляющей компоненты x_1^*

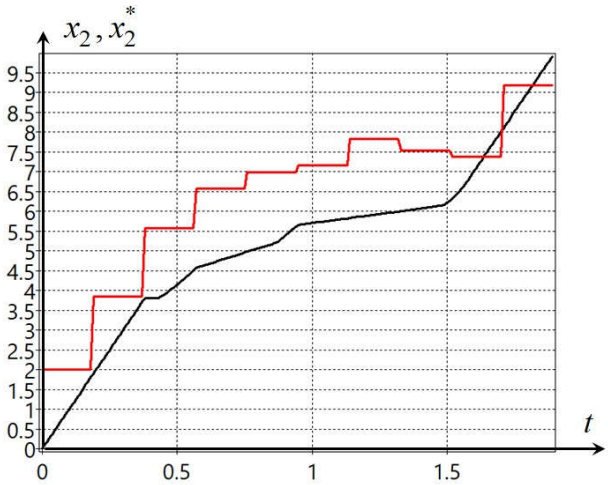

Рис. 4.16. График фазовой переменной x_2 и управляющей компоненты x_2^*

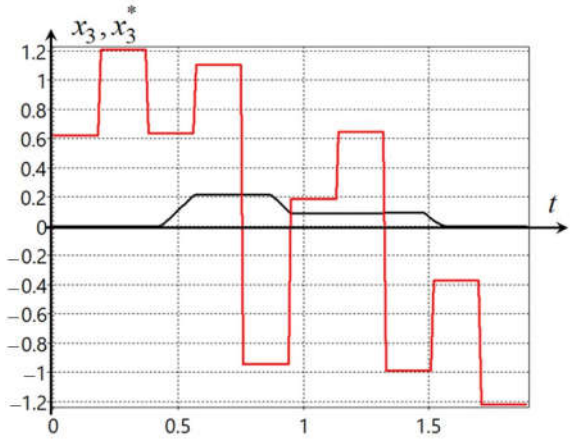

Рис. 4.17. График фазовой переменной x_3 и управляющей компоненты x_3^*

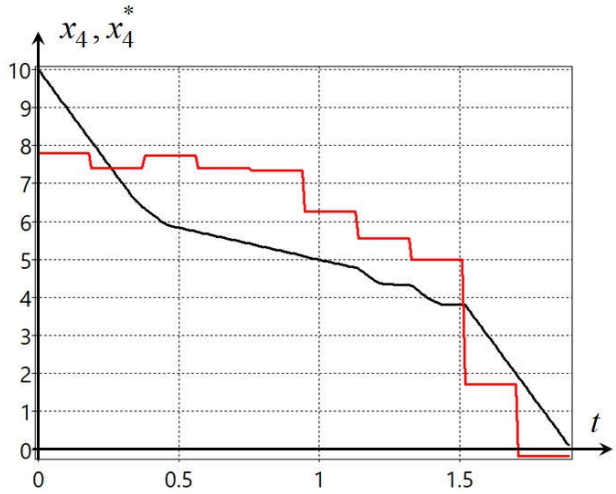

Рис. 4.18. График фазовой переменной x_4 и управляющей компоненты x_4^*

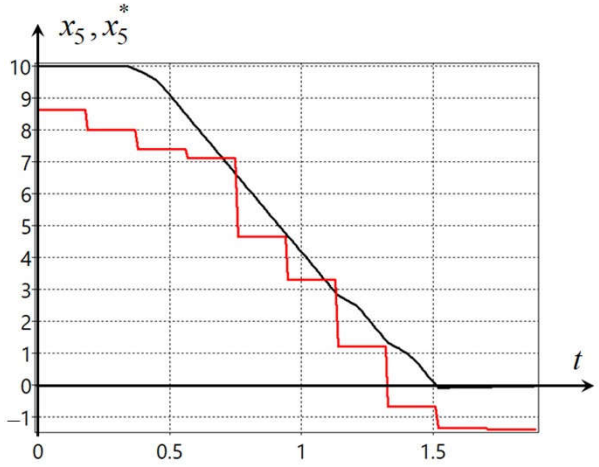

Рис. 4.19. График фазовой переменной x_5 и управляющей компоненты x_5^*

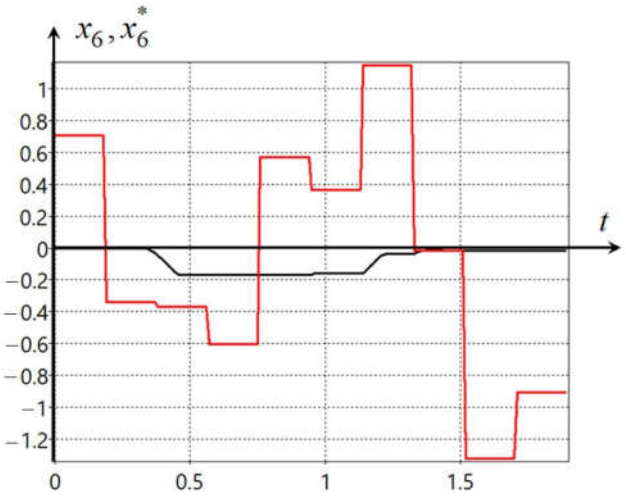

Рис. 4.20. График фазовой переменной x_6 и управляющей компоненты x_6^*

Далее на рисунках 4.21 – 4.28 представлены графики найденных оптимальных значений компонент вектора управления.

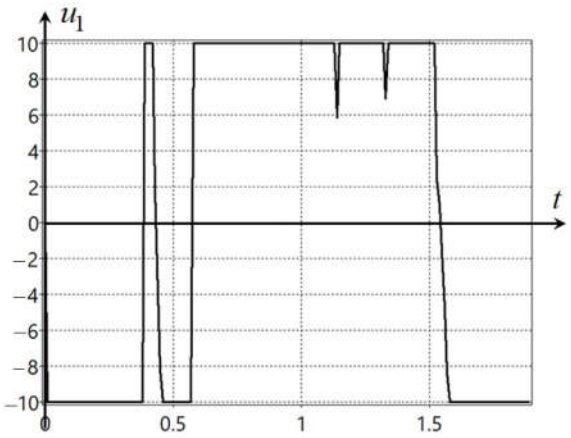

Рис. 4.21. График компоненты u_1 вектора управления

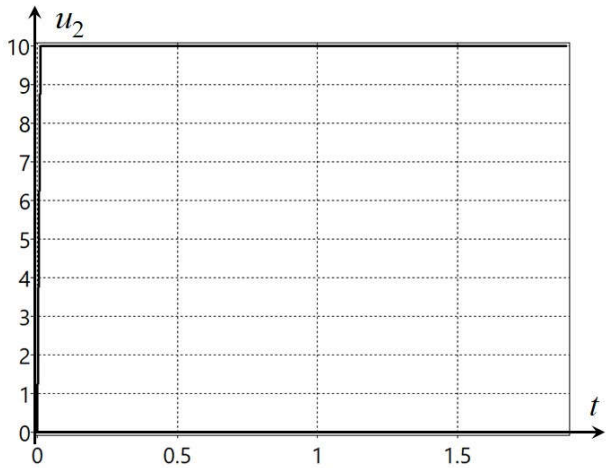

Рис. 4.22. График компоненты u_2 вектора управления

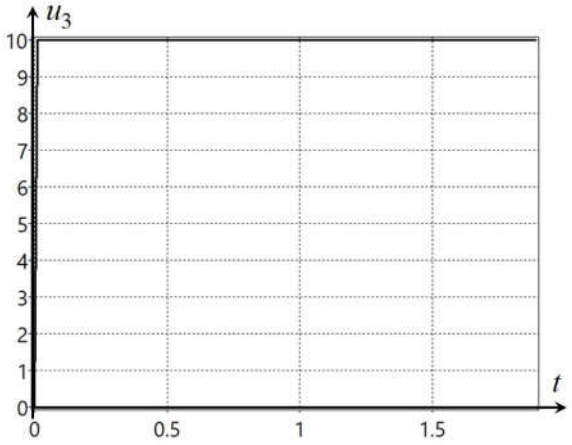

Рис. 4.23. График компоненты u_3 вектора управления

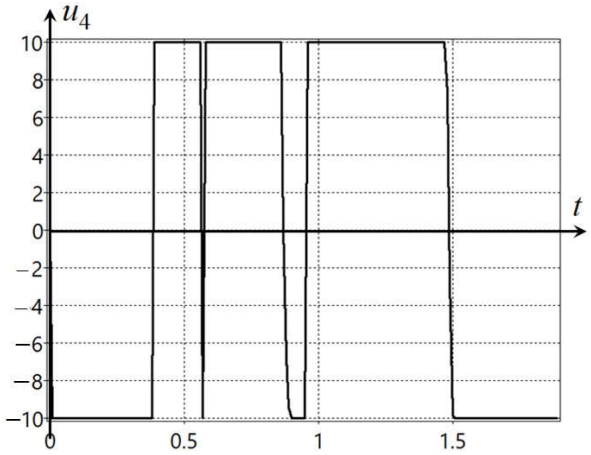

Рис. 4.24. График компоненты u_4 вектора управления

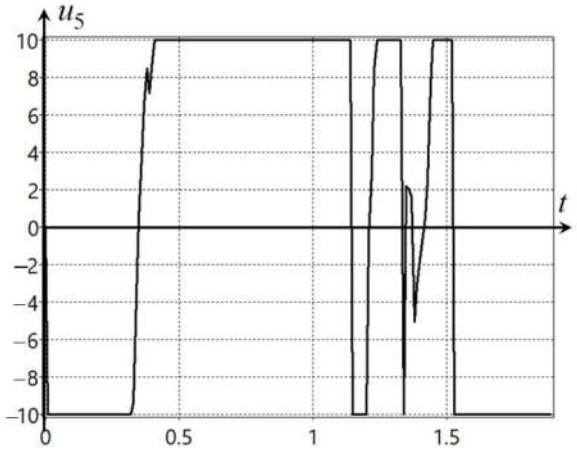

Рис. 4.25. График компоненты u_5 вектора управления

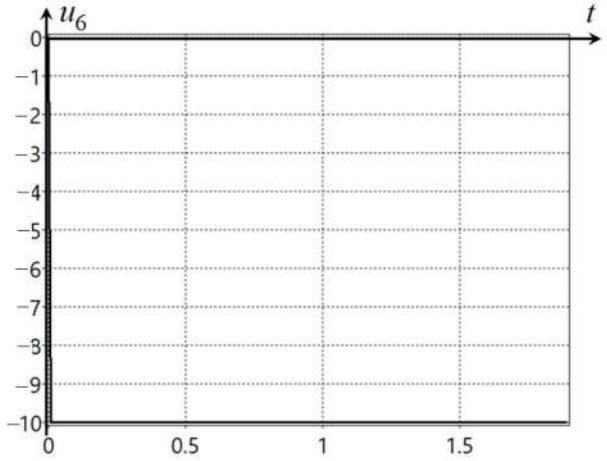

Рис. 4.26. График компоненты u_6 вектора управления

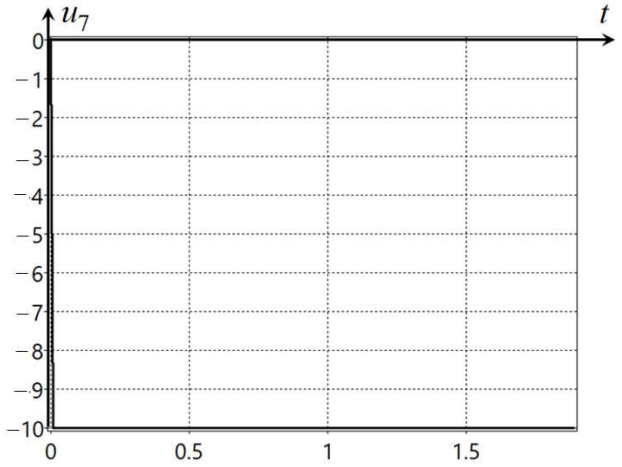

Рис. 4.27. График компоненты u_7 вектора управления

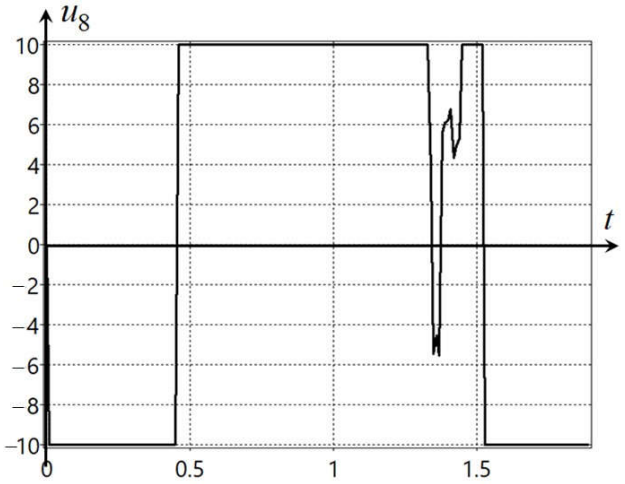

Рис. 4.28. График компоненты u_8 вектора управления

Как следует из приведенных графиков для управления роботом-меканум достаточно двух составляющих вектора управления. Две другие компоненты имеют предельные значения и не изменяются в процессе регулирования.

Действительно, как можно заметить, что в математической модели робота-меканум (4.22) вектор управления имел избыточную размерность $m > n$. Итак, компьютер сам нашел решение, как действовать в этом случае, и в найденном решении две компоненты управления, по сути, не участвуют в поиске оптимального решения. Это одна из ярких демонстраций интеллектуальной машинной автоматизации процесса создания систем управления.

Основным преимуществом представленного синтезированного подхода является его прикладной универсальный характер при решении задачи оптимального управления. Синтез системы стабилизации обеспечивает замкнутый контур управления, удовлетворяющий требованиям реализуемых систем управления. А универсальный подход к построению такого синтеза на основе современных методов символьного регрессии позволяет решать эту задачу численно на основе модели объекта и функционала независимо от их типа.

4.4. Задача управления квадрокоптером на основе адаптивного синтезированного оптимального управления

Задача управления, рассматриваемая в данном разделе, возникла в результате участия автора в составе команды ФИЦ ИУ РАН в соревнованиях по беспилотным летательным аппаратам АЭРОБОТ 2020. Соревнования показали важность формализации задач управления и разработки новых эффективных подходов к их решению.

Рис.4.29. Пример среды полета квадрокоптера со сложными фазовыми ограничениями

Рассмотрена задача оптимального управления квадрокоптером (см. рис.4.30) со сложными фазовыми ограничениями в виде статических препятствий в форме столбов и особых указанных зон, так называемых «окон», через которые объект должен обязательно пролететь. Пример такой среды представлен на рис.4.29 и является примером реальных условий с указанных соревнований.

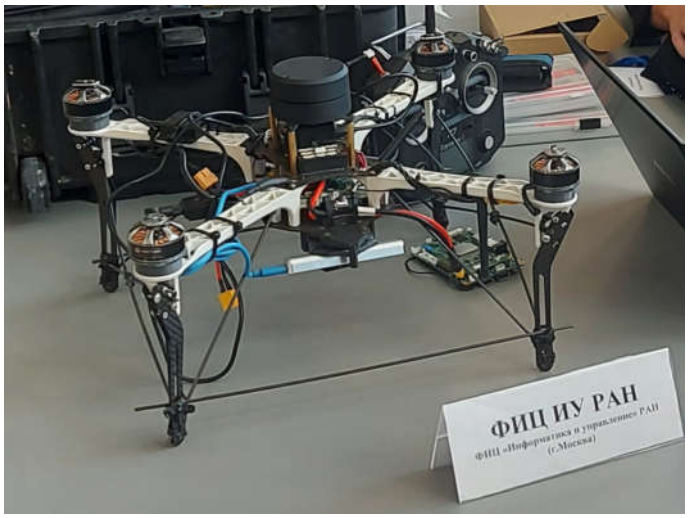

Рис.4.30. Квадрокоптер

Рассмотрим задачу оптимального управления пространственным движением квадрокоптера [234].

В общем случае математическая модель квадрокоптера как твердого тела имеет следующий вид:

$$\begin{aligned}
\ddot{x} &= F(\cos(\gamma)\sin(\theta)\cos(\psi) + \sin(\gamma)\sin(\psi))/m,\\
\ddot{y} &= Fcos(\gamma)cos(\theta)/m - g,\\
\ddot{z} &= F(\cos(\gamma)\sin(\theta)\sin(\psi) + \sin(\gamma)\cos(\psi))/m,\\
\ddot{\gamma} &= ((I_{yy} + I_{zz})\dot{\theta}\dot{\psi} + M_x)/I_{xx},\\
\ddot{\psi} &= ((I_{zz} + I_{xx})\dot{\gamma}\dot{\theta} + M_y)/I_{yy},\\
\ddot{\theta} &= ((I_{xx} + I_{yy})\dot{\gamma}\dot{\theta} + M_z/I_{zz},
\end{aligned} \tag{4.34}$$

где F - суммарная сила тяги всех винтов квадрокоптера, m - масса квадрокоптера, g - ускорение свободного падения, $g = 9{,}80665$, M_x, M_y, M_z - управляющие моменты, создаваемые винтами квадрокоптера вокруг соответствующих осей.

На рис. 4.31 представлена связь углов поворота квадрокоптера с его осями.

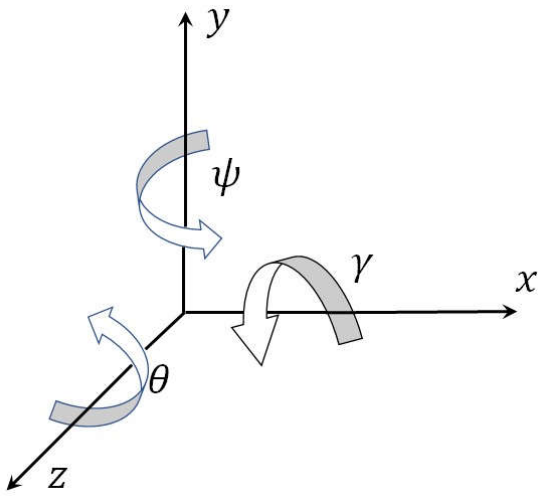

Рис.4.31. Система координат квадрокоптера

Для преобразования модели в векторную запись вводятся следующие обозначения: $x = x_1$, $y = x_2$, $z = x_3$, $\dot{x}_1 = x_4$, $\dot{x}_2 = x_5$, $\dot{x}_3 = x_6$, $\gamma = x_7$, $\psi = x_8$, $\theta = x_9$, $\dot{\gamma} = x_{10}$, $\dot{\psi} = x_{11}$, $\dot{\theta} = x_{12}$, $M_1 = M_x$, $M_2 = M_y$, $M_3 = M_z$.

В результате получается следующая математическая модель:

$$\begin{aligned}
\dot{x_1} &= x_4,\\
\dot{x_2} &= x_5,\\
\dot{x_3} &= x_6,\\
\dot{x_4} &= F(\cos(x_7)\,\sin(x_9)\cos(x_8) + \sin(x_7)\sin(x_8))/m,\\
\dot{x_5} &= F(\cos(x_7)\cos(x_9)/m - g,\\
\dot{x_6} &= F(\cos(x_7)\sin(x_9)\sin(x_8) + \sin(x_7)\cos(x_8))/m,\\
\dot{x_7} &= x_{10},\\
\dot{x_8} &= x_{11},\\
\dot{x_9} &= x_{12},\\
\dot{x_{10}} &= ((I_{yy} + I_{zz})x_{11}x_{12} + M_1)/I_{xx},\\
\dot{x_{11}} &= ((I_{zz} + I_{xx})x_{10}x_{12} + M_2)/I_{yy},\\
\dot{x_{12}} &= ((I_{xx} + I_{yy})x_{10}x_{11} + M_3)/I_{zz},
\end{aligned} \qquad (4.35)$$

где **x** - вектор пространства состояний, $\mathbf{x} = [x_1 \ldots x_n]^T$, M - вектор управляющих моментов, $\mathrm{M} = [M_1 \;\; M_2 \;\; M_3]^T$.

Как правило, квадрокоптеры, предлагаемые сегодня на рынке, выпускаются с предустановленной системой угловой стабилизации. Система стабилизации углов обеспечивает устойчивое положение квадрокоптера относительно заданных углов управляющими моментами:

$$M_i = w_i(x_7^* - x_7, x_8^* - x_8, x_9^* - x_9, x_{10}, x_{11}, x_{12}),\ i = 1,2,3. \qquad (4.36)$$

Предположим, что система угловой стабилизации достаточно быстро отрабатывает заданные углы квадрокоптера, по крайней мере, по сравнению с пространственным перемещением. В этом случае можно предположить, что управление пространственным перемещением квадрокоптера осуществляется с помощью его углового положения и силы тяги. Определим компоненты вектора пространственного управления: $x_7 = u_1$, $x_8 = u_2$, $x_9 = u_3$, $F/m = u_4$.

Тогда математическая модель пространственного движения квадрокоптера имеет следующий вид:

$$\dot{x}_1 = x_4,$$
$$\dot{x}_2 = x_5,$$
$$\dot{x}_3 = x_6, \tag{4.37}$$
$$\dot{x}_4 = u_4(\sin(u_3)\cos(u_2)\cos(u_1) + \sin(u_1)\sin(u_2)),$$
$$\dot{x}_5 = u_4\cos(u_3)\cos(u_1) - g_c,$$
$$\dot{x}_6 = u_4(\cos(u_2)\sin(u_1) - \cos(u_1)\sin(u_2)\sin(u_3)),$$

где g – ускорение свободного падения, $g = 9.80665$. Модель (4.37) описывает пространственное перемещение центра масс квадрокоптера за счет изменения его углов u_1, u_2, u_3 и суммарной тяги винтов u_4.

Управление имеет следующие ограничения:

$$-\frac{\pi}{12} \le u_1 \le \frac{\pi}{12}, \quad -\pi \le u_2 \le \pi,$$
$$-\frac{\pi}{12} \le u_3 \le \frac{\pi}{12}, \quad 0 \le u_4 \le 12. \tag{4.38}$$

Задано начальное положение квадрокоптера

$$\mathbf{x}^0 = [0\ \ 5\ \ 0\ \ 0\ \ 0\ \ 0]^T. \tag{4.39}$$

Пусть задано следующее терминальное состояние:

$$\mathbf{x}(t_f) = \mathbf{x}^f = [10\ \ \ 5\ \ \ 10\ \ \ 0\ \ 0\ \ 0]^T. \tag{4.40}$$

Фазовые ограничения заданы в виде цилиндров:

$$\varphi_i(\mathbf{x}) = r_i - \sqrt{\left(x_{\{1,i\}} - x_1\right)^2 + \left(x_{\{3,i\}} - x_3\right)^2} \le 0,\ i = 1,2, \tag{4.41}$$

где $r_1 = 2.5$, $r_2 = 2.5$, $x_{\{1,1\}} = 2.5$, $x_{\{1,2\}} = 7.5$, $x_{\{3,1\}} = 2.5$, $x_{\{3,2\}} = 7.5$.

В задаче зададим еще один тип фазовых ограничений в виде «окна»:

$$\eta(\mathbf{x}) = \sqrt{\sum_{i=1}^{3}(z_i - x_i)}, \tag{4.42}$$

где $z_1 = 5$, $z_2 = 5$, $z_3 = 5$.

Функционал качества задан в следующем виде:

$$J = t_f + p_1 \int_0^{t_f} \sum_{i=1}^{2} \vartheta\left(\varphi_i(\mathbf{x})\right) dt + p_2 \int_0^{t_f} \vartheta(\eta(\mathbf{x}))\, dt +$$
$$p_3 \left\|\mathbf{x}^f - \mathbf{x}(t_f)\right\| \to \min, \tag{4.43}$$

где $p_1 = 2,\ p_2 = 3,\ p_3 = 1$ – заданные весовые коэффициенты, t_f – время достижения терминального состояния:

$$t_f = \begin{cases} t, \text{если } t < t^+ \text{ и } \left\|\mathbf{x}^f - \mathbf{x}(t_f)\right\| \le \varepsilon, \\ t^+ - \text{ иначе,} \end{cases} \tag{4.44}$$

$t^+ = 5.6, \varepsilon = 0.01$.

Сначала решим сформулированную задачу оптимального управления в классической постановке, что требует нахождения функции управления как функции времени

$$\mathbf{u} = \mathbf{v}(t) \in \mathrm{U}.$$

Для этого используем прямой подход, состоящий в том, что задача оптимального управления преобразуется в конечномерную оптимизационную задачу нелинейного программирования. Этот подход содержит разбиение оси времени на интервалы $\mathrm{T} = (0, t_1, \dots, t_K)$ и далее нахождение в каждом интервале функции оптимального управления, как полиномиальной функции времени, зависящей от искомых параметров

$$\mathbf{v}^i(t) = q_{0,i} + q_{i,1}(t - t_{i-1}) + \cdots, \tag{4.45}$$

где $q_{0,i},\ q_{1,i}$ – искомые параметры, i – номер интервала, $i = 1, \dots, K$, K – количество интервалов.

Полагая, что функция управления непрерывна, то на каждом интервале число искомых параметров будет равно L_i, где L_i – порядок полинома в интервале i. Пусть во всех интервалах используются полиномиальные функции первого порядка $L_i = 1$. В этом случае каждый искомый параметр определяет значение управления на границе интервала. Если все интервалы одинаковы, то функция управления определяется уравнением:

$$v_j^i(t) = q_{i+(j-1)m} + \left(q_{i+(j-1)m+1} - q_{i+(j-1)m}\right)\frac{t - i\Delta t}{\Delta t}, \tag{4.46}$$

где $j = 1, \dots, m,\ i = 1, \dots, L$.

Итого в задаче необходимо найти вектор из $(L+1)m$ параметров

$$\mathbf{q} = \left[q_1 \dots q_{(L+1)m}\right]^T. \tag{4.47}$$

Функция управления находится в виде кусочно-линейной аппроксимации (4.46). При поиске был задан интервал времени $\Delta t = 0.4$, поэтому необходимо найти вектор параметров размерностью

$$\left(\frac{5.6}{0.4} + 1\right) 4 = 60.$$

Для решения задачи использовался эволюционный гибридный алгоритм [151]. В результате было получено следующее решение:

$$\mathbf{q} = [10.9213\ \ 3.9709\ \ -18.6827 - 5.5587$$
$$1.5187\ \ -13.5223\ \ -1.7807\ \ 0.0567$$
$$3.5280\ \ -0.8471\ \ 13.5166\ \ 1.8149$$
$$11.1227\ \ 18.7546\ \ 0.0509\ \ -3.2173$$
$$-7.8475\ \ 12.4403\ \ 10.6162\ \ -0.5559$$
$$-15.3702\ \ -11.1433\ \ 2.6074\ \ 13.3744$$
$$-1.3156\ \ -6.3763\ \ -1.6584\ \ 20.0000$$
$$-0.2657\ \ -0.1683\ \ -15.9205\ \ -1.1586$$
$$-7.7407\ \ -11.7280\ \ -0.0132\ \ 18.9735$$
$$-2.4755\ \ -0.9886\ \ 8.3376\ \ -0.7431$$
$$-2.2716\ \ -0.0943\ \ 15.6614\ \ 10.8643$$
$$15.8650\ \ 10.9646\ \ 19.2823\ \ 8.1150$$
$$8.1698\ \ 19.0514\ \ 4.0390\ \ 13.8275$$
$$5.6238\ \ 19.7189\ \ 18.8842\ \ 6.4341]^T\ .$$

Значение критерия составило $J = 5.7191$.

На рис. 4.32 представлена проекция найденной оптимальной траектории движения квадрокоптера на горизонтальную плоскость $\{x_1; x_3\}$. На рис. 4.32 красные сферы – фазовые ограничения, маленькая черная сфера – центр «окна».

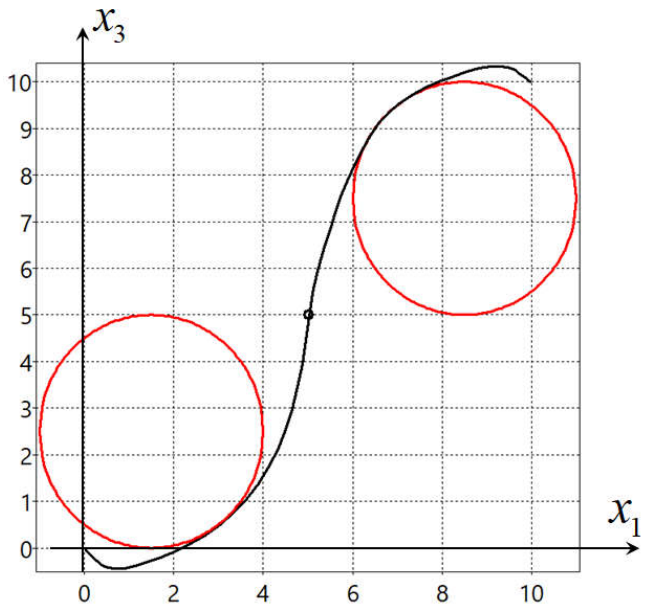

Рис.4.32. Прямое решение задачи оптимального управления

Далее задача решается на основе принципа синтезированного оптимального управления.

На первом этапе решается задача пространственной стабилизации объекта с целью получения устойчивой точки равновесия в пространстве состояний. Для этого снова воспользуемся методом сетевого оператора.

Для численного решения задачи начальная область для синтеза системы стабилизации берется в виде конечного числа точек в пространстве состояний. Для этого удобно задавать одно исходное состояние и отклонения от него:

$$\mathbf{x}^{0,j} = \mathbf{x}^0 - \mathbf{\Delta}_0 + 2 \odot (j)_2 \mathbf{\Delta}_0, \quad j = 1, \dots, 2^n - 1,$$

где $\mathbf{x}^0$ – заданное начальное состояние, $\mathbf{\Delta}_0$ – вектор отклонений, $\mathbf{\Delta}_0 = [\Delta_1 \dots \Delta_n]^T$, $\odot$ – Адамарово произведение, $(j)_2$ – двоичный код числа j.

Для построения множества начальных состояний был задан вектор отклонений $\Delta_0 = [2\ \ 2\ \ 2\ \ 0\ \ 0\ \ 0]^T$. Начальное и конечное состояния на этапе синтеза были равны $\mathbf{x}^0 = \mathbf{x}^f = [0\ \ 5\ \ 0\ \ 0\ \ 0\ \ 0]^T$.

В результате была получена следующая система стабилизации

$$u_i = \begin{cases} u_i^+, \text{ если } \bar{u}_i \geq u_i^+, \\ u_i^-, \text{ если } \bar{u}_i \geq u_i^-, \\ \bar{u}_i - \text{иначе}, \end{cases} \quad i = 1,2,3,4, \qquad (4.48)$$

где математические выражения для $\bar{u}_i$, $i = 1,2,3,4$, определяются как выходные элементы матрицы сетевого оператора, представленной на рис. 4.34 с параметрами $q_1 = 7.26733$, $q_2 = 11.46021$, $q_3 = 12.77271$, $q_4 = 3.20630$, $q_5 = 8.36914$, $q_6 = 5.50562$. Значения управлений в каждый момент времени вычисляются по матрице сетевого оператора на борту объекта.

Математические выражения функций управления, описываемых матрицей, представленной на рис. 4.34, имеют следующий сложный нелинейный вид:

$$\bar{u}_1 = \mu(C),$$

$$\bar{u}_2 = \bar{u}_1 - \bar{u}_1^3,$$

$$\bar{u}_3 = \bar{u}_2 + \rho_{19}(W + \mu(C)) + \rho_{17}(A),$$

$$\bar{u}_4 = \bar{u}_3 + \ln(|\bar{u}_2|) + \operatorname{sgn}\big(W + \mu(C)\big)\sqrt{|W + \mu(C)|} +$$

$$+\rho_{19}(W) + \arctan(H) + \operatorname{sgn}(F) + \arctan(E) + \exp(q_2(x_2^f - x_2)) + \sqrt{q_1},$$

где

$$C = q_6(x_6^f - x_6) + q_3(x_3^f - x_3),$$

$$W = V + \tanh(G) + \exp(D),$$

$$A = q_1(x_1^f - x_1) + q_4(x_4^f - x_4),$$

$$H = G + \tanh(F) + \rho_{18}(B),$$

$$F = E + C + \arctan(D) - B,$$

$$E = D + \operatorname{sgn}(x_5^f - x_5) + \left(x_2^f - x_2\right)^3,$$

$$V = exp(H) + cos(q_6(x_6^f - x_6)) + sgn(D)\sqrt{|D|},$$

$$G = F + \sqrt[3]{E} + \sin(A),$$

$$B = \sin(q_6(x_6^f - x_6)) + q_5(x_5^f - x_5) + q_2(x_2^f - x_2) + \cos(q_1) + \vartheta(x_2^f - x_2),$$

$$D = \rho_{17}(C) + B^3 + A + \vartheta(q_5(x_5^f - x_5)) + \left(x_5^f - x_5\right)^2,$$

$$\mu(z) = \begin{cases} z, & \text{если } |z| < 1, \\ \operatorname{sgn}(z) - \text{иначе}, \end{cases}$$

$$\rho_{17}(z) = \operatorname{sgn}(z)\ln(|z| + 1),$$

$$\rho_{18}(z) = \operatorname{sgn}(z)(\exp(|z|) - 1),$$

$$\rho_{19}(z) = \operatorname{sgn}(z)exp(-|z|).$$

На рис. 4.33 представлены траектории стабилизации квадрокоптера в начало координат в проекции на горизонтальную плоскость из восьми начальных состояний.

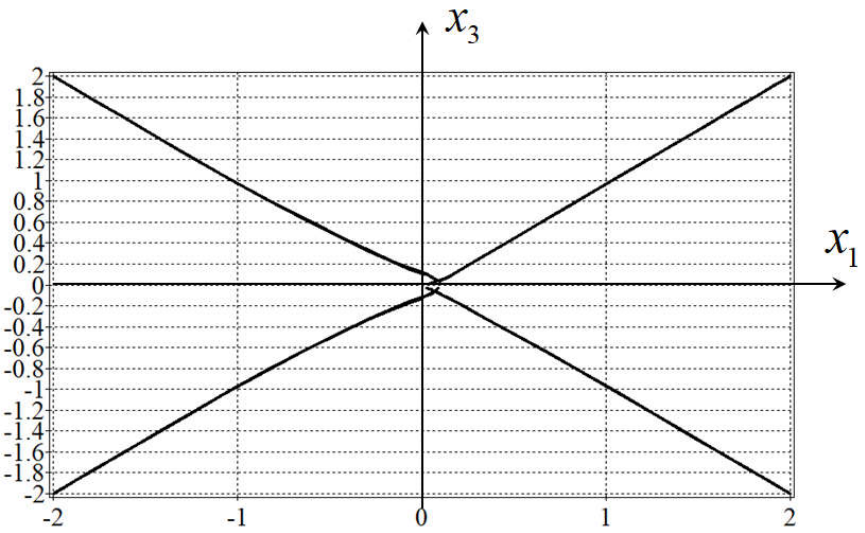

Рис.4.33. Проекция движения квадрокоптера на горизонтальную плоскость при стабилизации из разных начальных условий

На втором этапе находились положения точек стабилизации для оптимального управления объектом. Точки переключались с временным интервалом 0,4. На каждом интервале нужно было найти три координаты точки $\mathbf{x}^* = [x_1^* \; x_2^* \; x_3^*]^T$. Остальные координаты точек равны нулю. Так как число интервалов было $5.6/0.4 = 14$, то необходимо найти $14 \cdot 3 = 42$ параметра.

```
 PsiBasc:array [0..35,0..35] of integer=
((0,0,0,0,  0,0,0,0,  0,0,0,0,   1,0,0,0,  0,0,0,0, 0,0,0,0,  0,0,0,0,   0,0,0,0,  0,0,0,0),
 (0,0,0,0,  0,0,0,0,  0,0,0,0,   0,1,0,0,  0,0,0,9, 0,0,0,14, 0,0,0,0,   0,0,0,0,  0,0,0,0),
 (0,0,0,0,  0,0,0,0,  0,0,0,0,   0,0,1,0,  0,0,0,0, 0,0,0,0,  0,0,0,0,   0,0,0,0,  0,0,0,0),
 (0,0,0,0,  0,0,0,0,  0,0,0,0,   0,0,0,1,  0,0,0,0, 0,0,0,0,  0,0,0,0,   0,0,0,0,  0,0,0,0),

 (0,0,0,0,  0,0,0,0,  0,0,0,0,   0,0,0,0,  1,0,0,0, 0,2,0,10, 0,0,0,0,   0,0,0,0,  0,0,0,0),
 (0,0,0,0,  0,0,0,0,  0,0,0,0,   0,0,0,0,  0,1,0,0, 0,0,0,0,  0,0,0,0,   0,0,0,0,  0,0,0,0),
 (0,0,0,0,  0,0,0,0,  0,0,0,0,   1,0,0,0,  0,0,0,11, 0,0,0,0, 0,0,0,0,   0,0,0,0,  0,0,0,4),
 (0,0,0,0,  0,0,0,0,  0,0,0,0,   0,1,0,0,  0,0,0,0, 0,0,0,0,  0,0,0,0,   0,0,0,0,  0,0,0,0),

 (0,0,0,0,  0,0,0,0,  0,0,0,0,   0,0,1,0,  0,0,0,0, 0,0,0,0,  0,0,0,0,   0,0,0,0,  0,0,0,0),
 (0,0,0,0,  0,0,0,0,  0,0,0,0,   0,0,0,1,  0,0,0,0, 0,0,0,0,  0,0,0,0,   0,0,0,0,  0,0,0,0),
 (0,0,0,0,  0,0,0,0,  0,0,0,0,   0,0,0,0,  1,0,0,0, 0,0,0,0,  0,0,0,0,   0,0,0,0,  0,0,0,0),
 (0,0,0,0,  0,0,0,0,  0,0,0,0,   0,0,0,0,  0,1,0,0, 0,0,0,0,  0,0,0,0,   0,0,0,0,  0,0,0,0),

 (0,0,0,0,  0,0,0,0,  0,0,0,0,   2,0,0,0,  0,0,1,0, 0,0,0,0,  0,0,0,0,   0,0,0,0,  0,0,0,0),
 (0,0,0,0,  0,0,0,0,  0,0,0,0,   0,2,0,0,  0,0,0,1, 0,0,0,0,  0,0,0,0,   0,0,0,0,  0,0,0,6),
 (0,0,0,0,  0,0,0,0,  0,0,0,0,   0,0,2,0,  0,0,0,0, 1,0,0,0,  0,0,0,0,   0,0,0,0,  0,0,0,0),
 (0,0,0,0,  0,0,0,0,  0,0,0,0,   0,0,0,2,  0,0,1,0, 0,0,0,0,  0,0,0,0,   0,0,0,0,  0,0,0,0),

 (0,0,0,0,  0,0,0,0,  0,0,0,0,   0,0,0,0,  2,0,0,1, 0,9,0,0,  0,0,0,0,   0,0,0,0,  0,0,0,0),
 (0,0,0,0,  0,0,0,0,  0,0,0,0,   0,0,0,0,  0,2,0,12,1,0,0,0,  0,0,0,11,  0,0,0,0,  0,0,0,0),
 (0,0,0,0,  0,0,0,0,  0,0,0,0,   0,0,0,0,  0,0,1,0, 0,1,0,0,  0,12,0,0,  0,0,0,0,  0,0,17,0),
 (0,0,0,0,  0,0,0,0,  0,0,0,0,   0,0,0,0,  0,0,0,1, 0,14,0,0, 3,0,18,0,   0,0,0,0,  0,0,0,0),

 (0,0,0,0,  0,0,0,0,  0,0,0,0,   0,0,0,0,  0,0,0,0, 1,17,1,0, 0,0,0,0,   0,0,0,0,  0,0,0,0),
 (0,0,0,0,  0,0,0,0,  0,0,0,0,   0,0,0,0,  0,0,0,0, 0,1,0,1,  13,0,0,0,   0,6,0,0,  0,0,0,0),
 (0,0,0,0,  0,0,0,0,  0,0,0,0,   0,0,0,0,  0,0,0,0, 0,0,1,0,  1,0,0,0,   0,0,16,0,  0,0,0,0),
 (0,0,0,0,  0,0,0,0,  0,0,0,0,   0,0,0,0,  0,0,0,0, 0,0,0,1,  1,15,0,0,  0,0,0,0,  0,0,0,13),

 (0,0,0,0,  0,0,0,0,  0,0,0,0,   0,0,0,0,  0,0,0,0, 0,0,0,0,  1,1,8,0,   0,0,0,0,  0,0,0,10),
 (0,0,0,0,  0,0,0,0,  0,0,0,0,   0,0,0,0,  0,0,0,0, 0,0,0,0,  0,1,1,0,   4,8,0,0,  0,0,0,0),
 (0,0,0,0,  0,0,0,0,  0,0,0,0,   0,0,0,0,  0,0,0,0, 0,0,0,0,  0,0,1,6,   0,0,0,0,  0,0,0,13),
 (0,0,0,0,  0,0,0,0,  0,0,0,0,   0,0,0,0,  0,0,0,0, 0,0,0,0,  0,0,0,1,   1,0,0,0,  0,0,0,0),

 (0,0,0,0,  0,0,0,0,  0,0,0,0,   0,0,0,0,  0,0,0,0, 0,0,0,0,  0,0,0,0,   1,1,0,0,  0,0,0,0),
 (0,0,0,0,  0,0,0,0,  0,0,0,0,   0,0,0,0,  0,0,0,0, 0,0,0,0,  0,0,0,0,   0,1,1,0,  3,0,0,19),
 (0,0,0,0,  0,0,0,0,  0,0,0,0,   0,0,0,0,  0,0,0,0, 0,0,0,0,  0,0,0,0,   0,0,1,1,  0,0,19,0),
 (0,0,0,0,  0,0,0,0,  0,0,0,0,   0,0,0,0,  0,0,0,0, 0,0,0,0,  0,0,0,0,   0,0,0,1,  1,0,0,4),

 (0,0,0,0,  0,0,0,0,  0,0,0,0,   0,0,0,0,  0,0,0,0, 0,0,0,0,  0,0,0,0,   0,0,0,0,  1,23,0,0),
 (0,0,0,0,  0,0,0,0,  0,0,0,0,   0,0,0,0,  0,0,0,0, 0,0,0,0,  0,0,0,0,   0,0,0,0,  0,1,1,7),
 (0,0,0,0,  0,0,0,0,  0,0,0,0,   0,0,0,0,  0,0,0,0, 0,0,0,0,  0,0,0,0,   0,0,0,0,  0,0,1,1),
 (0,0,0,0,  0,0,0,0,  0,0,0,0,   0,0,0,0,  0,0,0,0, 0,0,0,0,  0,0,0,0,   0,0,0,0,  0,0,0,1));
```

Рис.4.34. Матрица сетевого оператора для пространственной стабилизации квадрокоптера

Для большей адаптивности разрабатываемой системы управления к возможным отклонениям начальных условий, что довольно часто случается с такими объектами, как квадрокоптер, на втором этапе принципа синтезированного оптимального управления дополнительно учитывали возможные неопределенности начальных условий. Такой подход дополнительно был назван принцип адаптивного синтезированного оптимального управления.

Поэтому при поиске параметров критерий качества учитывал набор начальных состояний, который определялся вариациями первых трех координат вектора состояния

$$x_i(0) = x_i^0 \pm 0.2, \quad i = 1,2,3.$$

Для поиска искомых координат использовался эволюционный гибридный алгоритм [151].

Гибридный алгоритм был предложен в результате проводимых исследований работы различных популяционных алгоритмов в задачах оптимального управления. В ходе экспериментов было замечено, что эволюционные алгоритмы работают быстрее, если при эволюционном преобразовании каждого возможного решения они используют как можно больше информации о значениях целевой функции в пространстве поиска. Алгоритм PSO при построении новых возможных решений использует информацию о текущем наилучшем решении, о наилучшем решении среди случайно выбранных информаторов и о предыдущих значениях целевой функции для каждого возможного решения. Алгоритм GWO использует информацию о некоторых текущих наилучших возможных решениях. Однако если целевая функция имеет сложный вид или включает множество сложных ограничений, то эти алгоритмы останавливаются в некоторых точках локального минимума. Генетический алгоритм в этих случаях начинает работать лучше, чем другие эволюционные алгоритмы. GA часто может сместить поиск от текущего локального минимума.

Разработанный гибридный алгоритм включает в себя все три перечисленных выше алгоритма. Изначально создаются все массивы и случайным образом выбирается тип эволюционного преобразования: GA, GWO или PSO. В каждом поколении количество преобразований каждого алгоритма примерно одинаково. Псевдокод алгоритма представлен в разделе 2.5.5 диссертации.

На рис.4.35 представлены проекции восьми траекторий квадрокоптера из восьми начальных состояний на горизонтальную плоскость. Черными квадратиками изображены проекции найденных управляющих точек равновесия.

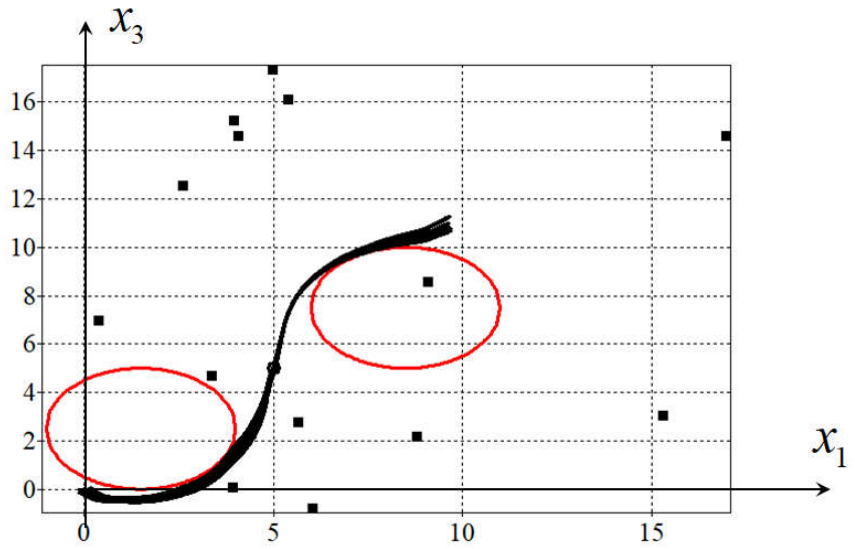

Рис.4.35. Проекция движения квадрокоптера на горизонтальную плоскость из разных начальных условий

Далее в таблице представлены экспериментальные результаты применения полученных законов управления на основе прямого и синтезированного подходов в условиях наличия неопределенностей начальных условий, представленных в виде аддитивных возмущений.

Таблица 4.4 содержит значения критерия качества для полученных решений при наличии возмущений. Возмущения представляли собой случайные изменения начальных условий из диапазона $\pm 0{,}2$. В первом столбце указан номер эксперимента, второй столбец – значение функционала при адаптивном синтезированном управлении, третий – при прямом подходе. Внизу таблицы посчитано среднее значение функционала по проведенным испытаниям, а также среднеквадратичное отклонение.

Таблица 4.4: Значения функционала качества для полученных решений в присутствии возмущений в начальных условиях

№ эксперимента	Синтезированное управление	Кусочно-линейная аппроксимация
1	6.8426	13.2417
2	6.5790	10.8158
3	6.8764	9.8738
4	6.5982	16.4428
5	6.9731	11.1727
6	6.5661	8.7628
7	6.5809	13.4681
8	6.9493	13.2927
9	6.4154	16.3510
10	6.7282	10.5320
Среднее	6.7106	12.3948
Ср.кв.откл.	0.19031	2.61878

Как видно из таблицы, управление, рассчитанное на основе принципа адаптивного синтезированного управления значительно менее чувствительно к возмущениям, чем на основе прямого подхода. Несмотря на то, что при адаптивном синтезированном подходе значение функционала качества немного хуже, чем при расчетном оптимальном управлении $J = 5.7191$, можно сказать, что в данном случае это целенаправленная жертва для обеспечения стабильной работы (среднеквадратичное отклонение функционала всего 0.19031) при возможных неопределенностях начальных условий.

После отработки технологии расчета управления для участка пути решалась задача прохождения квадрокоптера по маршруту. На маршруте есть окна, через которые должен обязательно пролететь квадрокоптер, и препятствия в виде столбов, с которыми квадрокоптер не должен столкнуться.

Система стабилизации, полученная на первом этапе алгоритма, остается той же. На втором этапе рассматривается следующая задача оптимального управления.

Имеется математическая модель (4.37) с введенной системой стабилизации (4.48). Положение старта совпадает с положением финиша

$$\mathbf{x}^0 = \mathbf{x}^f = [0\ \ 5\ \ 0\ \ 0\ \ 0\ \ 0]^T.$$

Необходимо найти управление в виде множества точек в пространстве состояний, минимизируя следующий критерий качества:

$$J_{route} = t_f + p_1\left\|\mathbf{x}^f - \mathbf{x}(t_f)\right\| + p_2\sum_{i=0}^{N}\int_0^{t_f}\vartheta\big(\varphi_i(\mathbf{x})\big)dt + p_3\sum_{j=1}^{S}\vartheta\left(\min_t \delta_j(\mathbf{x}) - \varepsilon\right) \to \min_{\mathbf{x}^*}, \quad (4.49)$$

где $p_1 = 2, p_2 = 3, p_3 = 3,$

$$\varphi_i(\mathbf{x}) = r_i - \sqrt{\left(x_{i,1} - x_1\right)^2 + \left(x_{i,3} - x_3\right)^2}, i = 1, \dots, N = 4, \quad (4.50)$$

$r_1 = r_2 = r_3 = r_4 = 2,\ x_{1,1} = 5,\ x_{1,3} = 0,\ x_{2,1} = 10,\ x_{2,3} = 5,\ x_{3,1} = 5,\ x_{3,3} = 10,$ $x_{4,1} = 0, x_{4,3} = 5,$

$$\delta_j(\mathbf{x}) = \sqrt{\left(y_{i,1} - x_1\right)^2 + \left(y_{j,3} - x_3\right)^2}, j = 1, \dots, S = 7, \quad (4.51)$$

$y_{1,1} = 5, \quad y_{1,3} = -2, \quad y_{2,1} = 10, \quad y_{2,3} = 0, \quad y_{3,1} = 12, \quad y_{3,3} = 5,$ $y_{4,1} = 10,\ y_{4,3} = 10,\ y_{5,1} = 5,\ y_{5,3} = 12,\ \ y_{6,1} = 0,\ y_{6,3} = 10,\ y_{7,1} = 2,\ y_{7,3} = 5,$ $\varepsilon = 0.6.$

Конечное время t_f определяется уравнением (4.44) с $t^+ = 14.4,\ \varepsilon_0 = 0.1.$ Необходимо найти координаты управляющих точек на каждом интервале времени, $\Delta t = 0.8.$ Искомый вектор включает в себя $3M$ параметров, где

$$M = \left\lfloor\frac{t^+}{\Delta t}\right\rfloor = \left\lfloor\frac{14.4}{0.8}\right\rfloor = 18,$$

то есть необходимо найти 54 параметра.

С помощь гибридного эволюционного алгоритма было найдено следующее решение:

$$\begin{gathered}
\mathbf{x}^{*,1} = [4.83910 \quad 1.14025 \quad -5.22899]^T, \\
\mathbf{x}^{*,2} = [11.07056 \quad 6.79389 \quad -2.48647]^T, \\
\mathbf{x}^{*,3} = [9.19808 \quad 1.54674 \quad 15.87195]^T, \\
\mathbf{x}^{*,4} = [-0.12204 \quad 0.12276 \quad -1.82381]^T, \\
\mathbf{x}^{*,5} = [-4.08347 \quad 2.93658 \quad 5.89553]^T, \\
\mathbf{x}^{*,6} = [16.72896 \quad 2.18022 \quad 2.27907]^T, \\
\mathbf{x}^{*,7} = [1.18106 \quad 2.56582 \quad 14.41088]^T, \\
\mathbf{x}^{*,8} = [8.67198 \quad 5.78737 \quad -2.90409]^T, \\
\mathbf{x}^{*,9} = [8.59478 \quad 2.73948 \quad 11.33252]^T, \\
\mathbf{x}^{*,10} = [-1.25924 \quad -1.97448 \quad -1.42747]^T, \\
\mathbf{x}^{*,11} = [2.45445 \quad 7.42257 \quad -0.38164]^T, \\
\mathbf{x}^{*,12} = [8.68306 \quad -0.78496 \quad 15.41667]^T, \\
\mathbf{x}^{*,13} = [0.60972 \quad 7.02724 \quad 7.66403]^T, \\
\mathbf{x}^{*,14} = [-0.59975 \quad 0.39324 \quad -1.31307]^T, \\
\mathbf{x}^{*,15} = [-2.39004 \quad 7.95279 \quad 3.02003]^T, \\
\mathbf{x}^{*,16} = [2.52642 \quad 6.69332 \quad 9.17356]^T \\
\mathbf{x}^{*,17} = [-0.95896 \quad 4.42529 \quad -0.36318]^T, \\
\mathbf{x}^{*,18} = [-0.01193 \quad 5.02821 \quad 15.40007]^T.
\end{gathered} \tag{4.52}$$

Значение функционала для найденного решения составило $J_{route} = 14{,}7010$. На рис. 4.36 представлены проекции полученной траектории при найденном управлении на горизонтальную плоскость $\{x_1;\ x_3\}$.

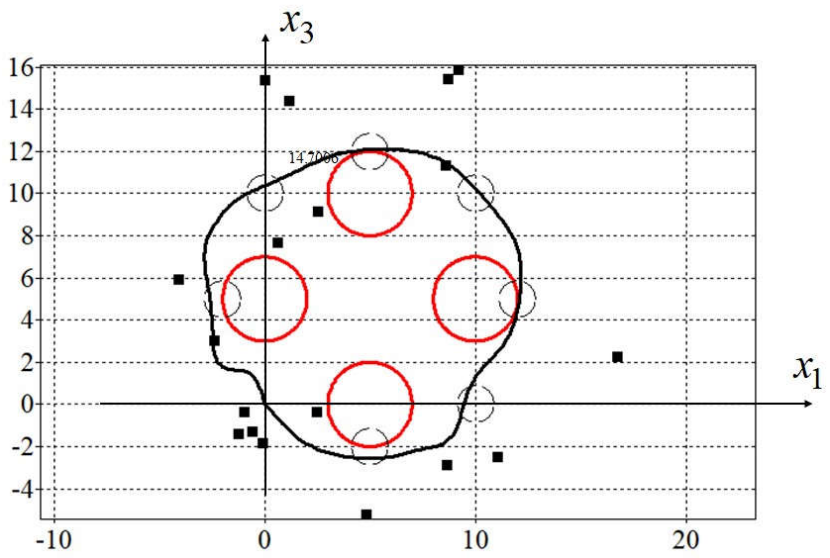

Рис.4.36. Траектория движения квадрокоптера при синтезированном оптимальном управлении

На рисунке красными окружностями обозначены фазовые ограничения, описываемые (4.50), небольшими черными окружностями представлены окна, описываемые (4.51), а маленькие черные прямоугольники – это найденные управляющие точки стабилизации (4.52). Как видно из представленного графика, квадрокоптер прекрасно справляется с поставленной задачей, пролетает все необходимые зоны и не сталкивается с фазовыми ограничениями.

4.5. Задача группового взаимодействия квадрокоптеров

Рассмотрим задачу оптимального управления групповым взаимодействием квадрокоптеров при выполнении общей задачи в пространстве с фазовыми ограничениями.

Пусть имеется группа из трех квадрокоптеров. Задача состоит в нахождении такого управления, которое обеспечит перемещение квадрокоптерами груза на гибких тягах из одной точки пространства в другую, не задев препятствий, при этом вес груза не позволяет выполнить задачу одним квадрокоптером. На пути перемещения груза могут размещаться некоторые препятствия, положение которых известно заранее.

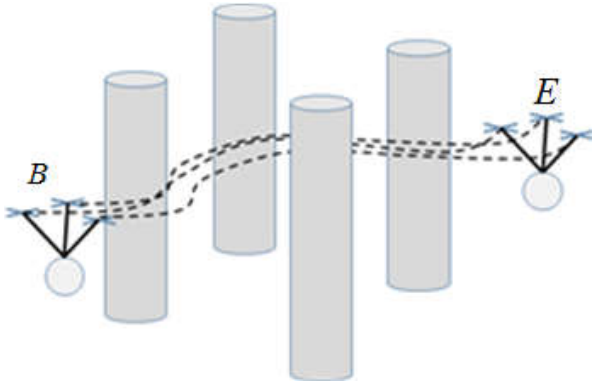

Рис. 4.37. Схема взаимодействия трех квадрокоптеров в группе: B, E - начальное и конечное положения квадрокоптеров; цилиндрами обозначены препятствия; пунктирными линиями - возможные траектории движения квадрокоптеров; шар изображает груз

Для эффективного решения поставленной задачи необходимо учитывать групповое взаимодействие объектов управления. Формально в рассматриваемой задаче заданы математические модели объектов управления, ограничения на управление, фазовые ограничения, начальное терминальное состояния, определяющие выполнение работы, а также задан некоторый критерий качества управления. Критерий качества управления определяется скоростью доставки груза до пункта назначения. Квадрокоптеры также не должны сталкиваться друг с другом, должны избегать столкновения с препятствиями и находиться на определенном расстоянии от груза, чтобы участвовать в допустимом распределении нагрузки по всем трем квадрокоптерам. Необходимо построить систему управления группой летающих роботов для достижения цели управления с оптимальным значением заданного критерия качества. Полученная система управления должна также обладать свойствами малой чувствительности к неопределенностям математических моделей объектов управления.

Представим решение задачи на основе принципа синтезированного оптимального управления.

Рассматриваем квадрокоптеры одного типа: симметричные с четырьмя винтами [235]. Математическую модель квадрокоптера задает следующая система дифференциальных уравнений:

$$\begin{gathered}
\dot{x}_1^j = x_4^j + \left(x_5^j \sin x_1^j + x_6^j \cos x_1^j\right) \sin x_2^j / \cos x_2^j, \\
\dot{x}_2^j = \left(x_5^j \sin x_1^j + x_6^j \cos x_1^j\right) / \cos x_2^j, \\
\dot{x}_3^j = x_5^j \sin x_1^j + x_6^j \cos x_1^j, \\
\dot{x}_4^j = x_5^j x_6^j (I_2 - I_3)/I_1 + M_1^j / I_1, \\
\dot{x}_5^j = x_4^j x_6^j (I_3 - I_1)/I_2 + M_2^j / I_2, \\
\dot{x}_6^j = x_4^j x_5^j (I_1 - I_2)/I_3 + M_3^j / I_3, \\
\dot{x}_7^j = x_{10}^j, \\
\dot{x}_8^j = x_{11}^j, \\
\dot{x}_9^j = x_{12}^j, \\
\dot{x}_{10}^j = F^j (\sin x_3^j \cos x_2^j \cos x_1^j + \sin x_1^j \sin x_2^j) - w_1^j, \\
\dot{x}_{11}^j = F^j \cos x_3^j \cos x_2^j \cos x_1^j - g - w_2^j, \\
\dot{x}_{12}^j = F^j (\sin x_1^j \cos x_2^j - \cos x_1^j \sin x_2^j \sin x_3^j) - w_3^j,
\end{gathered} \tag{4.53}$$

где j – номер квадрокоптера; w_k^j – компоненты весовой нагрузки квадрокоптера j от веса переносимого груза, $k = 1,2,3$; w_1^j, w_2^j, w_3^j – проекции нагрузки на оси x_7, x_8, x_9 соответственно; x_1^j, x_3^j – углы поворота квадрокоптера j вокруг горизонтальных осей; x_2^j – угол поворота квадрокоптера j вокруг вертикальной оси; x_4^j и x_6^j x_6^j угловые скорости вращения квадрокоптера j вокруг горизонтальных осей; x_5^j – угловая скорость вращения квадрокоптера j вокруг вертикальной оси; x_7^j, x_9^j – координаты центра масс квадрокоптера j на горизонтальной плоскости; x_8^j – высота квадрокоптераj; x_{10}^j, x_{11}^j, x_{12}^j – проекции линейной скорости квадрокоптера j на соответствующие оси; M_i^j – управляющие моменты, создаваемые винтами квадрокоптера j вокруг осей x_i, $i = 1,2,3$; F^j – общая тяга всех четырех винтов квадрокоптера с учетом поправки на массуm; g-

постоянная величина ускорения свободного падения; I_i- моменты инерции квадрокоптера вокруг осей $x_i, i = 1,2,3$.

Положение груза определяем вектором $\mathbf{y} \in \mathbb{R}^3$ состояния центра масс груза в подпространстве $\{x_7,\ x_8,\ x_9\}$

$$\mathbf{y} = [y_1\ \ y_2\ \ y_3]^T, \tag{4.54}$$

где y_1, y_2, y_3 - положения центра масс груза по осям x_7, x_8, x_9 соответственно.

Задано начальное и терминальное положения груза

$$\mathbf{y} = [y_1^0\ \ y_2^0\ \ y_3^0]^T, \tag{4.55}$$

$$\mathbf{y} = [y_1^f\ \ y_2^f\ \ y_3^f]^T. \tag{4.56}$$

Полагаем, что все три квадрокоптера одинаковы, а тяги, за которые они перемещают груз, имеют одинаковую длину: $d_1 = d_2 = d_3 = d$.

Для равномерного распределения нагрузки на квадрокоптеры все три квадрокоптера всегда должны находиться на одной высоте и быть расположены в вершинах равностороннего треугольника. Размер этого треугольника определяет высоту груза. Пусть размер треугольника определяет радиус описанной окружности R_0. Тогда разность высот квадрокоптера x_8^j, $j = 1,2,3$ и груза y_2 определяется как

$$x_8^j - y_2 = \sqrt{d^2 - R_0^2}, j = 1,2,3. \tag{4.57}$$

Определим проекции силы реакции нагрузки на каждый квадрокоптер (см. рис.4.38):

$$\mathbf{w} = \mathbf{w}^1 + \mathbf{w}^2 + \mathbf{w}^3. \tag{4.57}$$

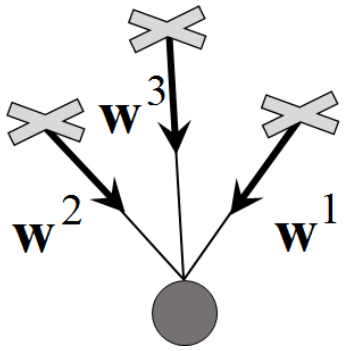

Рис. 4.38. Силы реакции нагрузки, действующие на квадрокоптеры

Для того, чтобы определить проекции указанных сил реакции, свяжем с равносторонним треугольником, который образуют квадрокоптеры, дополнительную систему координат (см. рис. 4.39).

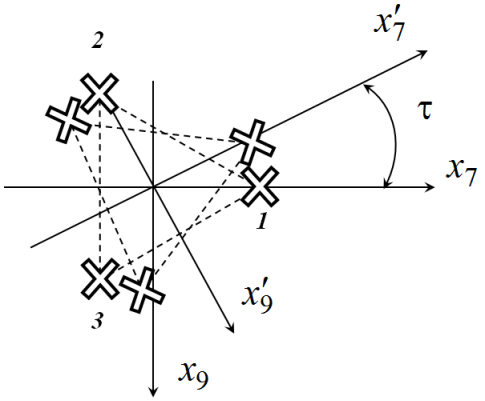

Рис. 4.39. Расположение квадрокоптеров в равностороннем треугольнике

Для расчета проекций сил реакций необходимо знать угол поворота τ равностороннего треугольника, образованного квадрокоптерами, относительно инерциальной системы координат. Полученные уравнения имеют вид

$$w_1^1 = -|w^1|R_0\cos(\tau)/d,$$
$$w_2^1 = |w^1|\sqrt{d^2 - R_0^2}/d, \quad (4.58)$$
$$w_3^1 = |w^1|R_0\sin(\tau)/d,$$

$$w_1^2 = |w^2|R_0(\cos(\tau) + \sqrt{3}\sin(\tau))/2d,$$
$$w_2^2 = |w^2|\sqrt{d^2 - R_0^2}/d, \quad (5.59)$$
$$w_1^2 = |w^2|R_0(\sin(\tau) - \sqrt{3}\cos(\tau))/2d,$$

$$w_1^3 = |w^3|R_0(\cos(\tau) - \sqrt{3}\sin(\tau))/2d,$$
$$w_2^3 = |w^3|\sqrt{d^2 - R_0^2}/d, \quad (4.60)$$
$$w_1^3 = |w^3|R_0(\sin(\tau) + \sqrt{3}\cos(\tau))/2d,$$

где

$$|w^i| = \sqrt{(w_1^i)^2 - (w_2^i)^2 + (w_3^i)^2}, i = 1,2,3\ , \quad (4.61)$$

$$R_0 = \sqrt{3((x_7^1 - x_7^2)^2 + (x_9^1 - x_9^2)^2)}/3, \quad (4.62)$$

$$\tau = \arctan\left(\frac{x_9^1 - (x_9^1 + x_9^2 + x_9^3)/3}{x_7^1 - (x_7^1 + x_7^2 + x_7^3)/3}\right). \quad (4.63)$$

Дополнительные фазовые ограничения в задаче оптимального управления определяются расположением квадрокоптеров в вершинах равностороннего треугольника на одной высоте и имеют вид

$$\chi_1 = |(x_7^1 - x_7^2)^2 + (x_9^1 - x_9^2)^2 - (x_7^1 - x_7^3)^2 - (x_9^1 - x_9^3)^2| - \varepsilon_p \le 0,$$
$$\chi_2 = |(x_7^1 - x_7^2)^2 + (x_9^1 - x_9^2)^2 - (x_7^2 - x_7^3)^2 - (x_9^2 - x_9^3)^2| - \varepsilon_p \le 0, \quad (4.64)$$
$$\chi_3 = |(x_8^1 - x_8^2)^2 - (x_8^1 - x_8^3)^2| - \varepsilon_p \le 0,$$
$$\chi_4 = |(x_8^1 - x_8^2)^2 - (x_8^2 - x_8^3)^2| - \varepsilon_p \le 0,$$

где ε_p –заданная малая положительная величина.

Координаты груза определяем из уравнений

$$y_1 = x_7^1 - \sqrt{L_{1,2}/3}\cos(\Omega),$$
$$y_2 = x_8^1 - \sqrt{d^2 - L_{1,2}/3}, \quad (4.65)$$
$$y_3 = x_9^1 - (x_7^1 - \sqrt{L_{1,2}/3}\sin(\Omega)),$$

где $\Omega = \arctan(|x_7^2 - x_7^3|/|x_9^2 - x_9^3|)$,$L_{1,2} = (x_7^1 - x_7^2)^2 + (x_9^1 - x_9^2)^2$.

Расстояние между каждым квадрокоптером и грузом не может быть больше и не должно быть значительно меньше длины тяги d. Данные ограничения формализуем в виде следующей системы неравенств:

$$\Phi_i(x^i) = L - d \le 0, \quad (4.66)$$
$$\Phi_{3+i}(x^i) = d - L - \varepsilon_r \le 0, \quad (4.67)$$

где $L = \sqrt{(x_7^i - y_1)^2 + (x_8^i - y_2)^2 + (x_9^i - y_3)^2}$, $i = 1,2,3$: ε_r – заданная малая положительная величина.

В области перемещения грузов существуют препятствия, которые определяют дополнительные фазовые ограничения:

$$\Phi_{6+(i-1)s+j}(x^i) = r_j - \sqrt{(x_7^i - x_{1,j})^2 + (x_9^i - x_{2,j})^2} \le 0,. \quad (4.68)$$

где $i = 1,2,3, j = 1,\dots,s$; s – количество препятствий; r_j – параметр препятствияj, определяющий размер препятствия; $x_{1,j}$, $x_{2,j}$ – координаты геометрического центра препятствия на плоскости $\{x_7, x_9\}$.

Далее запишем условия столкновения груза и тяг с препятствиями.

$$\chi_{4+j} = r_j - \sqrt{(y_1 - x_{1,j})^2 + (y_3 - x_{2,j})^2} \le 0, \tag{4.69}$$

$$\chi_{4+is+j} = \sqrt{(x_{1,j} - \tilde{x}_7^i)^2 + (x_{2,j} - \tilde{x}_9^i)^2} - r_j \le 0, \tag{4.70}$$

где$j = 1, \dots, s, i = 1,2,3.$

Уравнение (4.70) включает в себя величины $(\tilde{x}_7^i, \tilde{x}_9^i)$ – координаты точки пересечения проекции тяги на плоскость $\{x_7, x_9\}$ и проекции перпендикуляра от центра препятствия к линии тяги. Для вычисления этих координат используем следующие отношения:

если $\left|x_7^i - y_1\right| > \delta$ и$\left|x_9^i - y_3\right| > \delta$, где δ- небольшое положительное значение, то

$$\tilde{x}_7^i = \frac{(x_7^i - y_1)(x_9^i - y_3)}{(x_9^i - y_3)^2 + (x_7^i - y_1)^2}\left(\frac{(x_9^i - y_3)}{(x_7^i - y_1)} x_7^i + y_3 + x_{2,j} + \frac{(x_7^i - y_1)}{(x_9^i - y_3)} x_{1,j}\right), \tag{4.71}$$

$$\tilde{x}_9^i = \frac{(x_9^i - y_3)}{(x_7^i - y_1)} \tilde{x}_7^i - \frac{(x_9^i - y_3)}{(x_7^i - y_1)} x_7^i + y_3, \tag{4.72}$$

иначе,

если $\left|x_7^i - y_1\right| < \delta$ и$\left|x_9^i - y_3\right| > \delta$, то

$$\tilde{x}_7^i = x_7^i,\ \tilde{x}_9^i = x_{2,j}\ , \tag{4.73}$$

иначе,

если $\left|x_7^i - y_1\right| > \delta$ и$\left|x_9^i - y_3\right| < \delta$, то

$$\tilde{x}_7^i = x_{1,j}, \tilde{x}_9^i = x_9^i,. \tag{4.74}$$

Здесь$j = 1, \dots, s, i = 1,2,3.$

Заметим, что условия $\left|x_7^i - y_1\right| < \delta$ и $\left|x_9^i - y_3\right| < \delta$ никогда одновременно не выполняются.

Уравнения (4.69) - (4.74) формализуют условия отсутствия столкновения тяг, на которых перемещается груз, с препятствиями. Геометрический смысл используемых переменных объясняет рис. 4.40.

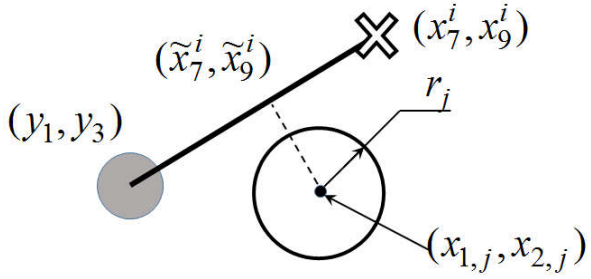

Рис. 4.40. Расположение тяги квадрокоптера и фазовое ограничение

Заметим, что фазовые ограничения, которые вытекают из условия отсутствия столкновений между роботами, учитывать в данной постановке задачи нет необходимости, поскольку эти ограничения поглощены ограничениями, связанными с условиями группового взаимодействия: квадрокоптеры должны находиться в вершинах равностороннего треугольника (4.64).

Таким образом, всего в задаче имеется $4 + 4s$ фазовых ограничений, определяющих групповое взаимодействие квадрокоптеров, и $6 + 3s$ индивидуальных фазовых ограничений для каждого квадрокоптера.

В качестве критерия качества управления используем аддитивную свертку времени доставки груза в терминальное положение и точности доставки этого груза в конечную точку (4.49). Также функционал качества управления включает в себя условия ненарушения фазовых ограничений в виде штрафных функций:

$$J = t_f + \alpha_1 \sqrt{\sum_{i=1}^{3} (y_i(t_f) - y_i^f)^2} + \alpha_2 \int_0^{t_f} \sum_{j=1}^{4+4s} \vartheta(\chi_j(x^1, x^2, x^3))dt +$$

$$+\alpha_3 \int_0^{t_f} (\sum_{i=1}^{3} (\vartheta\left(\Phi_i\left(x^i\right)\right) + \vartheta\left(\Phi_{3+i}\left(x^i\right)\right) +$$

$$+ \sum_{j=1}^{s} \vartheta(\Phi_{6+(i-1)s+j}(x^i)))dt \to \min_{\mathbf{u}^j(\cdot), j=1,2,3} \qquad (4.75)$$

где α_i – весовые коэффициенты, $i = 1,2,3$, $\mathbf{u}^j(\cdot) = [M_1^j(\cdot) \quad M_2^j(\cdot) \quad M_3^j(\cdot) \quad F^j(\cdot)]^T, j = 1,2,3$; $\vartheta(A)$ – функция Хэвисайда

$$\vartheta(A) = \begin{cases} 1, \text{если } A > 0, \\ 0 - \text{иначе}, \end{cases}$$

$$t_f = \begin{cases} t, \text{если } t < t^+ \text{ и } L_c(t) \le \varepsilon_f, \\ t^+ - \text{ иначе}, \end{cases}$$

$$L_c(t) = \sqrt{\sum_{j=1}^{3} \left(y_i(t) - y_i^f\right)^2}.$$

Согласно принципу синтезированного оптимального управления, на первом этапе решаем задачу синтеза стабилизирующего управления, обеспечивающего устойчивость объекта управления относительно некоторой заданной точки стабилизации в пространстве состояний. На втором этапе осуществляется поиск координат нескольких точек стабилизации с использованием методов конечномерной оптимизации. Управление осуществляется путем переключения точек стабилизации от одной к другой.

Для синтеза системы стабилизации рассмотрим модель (4.53) одного квадрокоптера $j = 1$ без груза $w_k^1 = 0,\ k = 1,2,3$ и без фазовых ограничений (4.64) - (4.74). Задаем набор начальных условий и одно терминальное условие. Нужно найти такую функцию управления, которая обеспечивает перемещение квадрокоптера из любых начальных условий в терминальное за минимальное время.

В результате синтеза системы стабилизации методом сетевого оператора было получены функции управления обратной связи для каждой компоненты вектора управления $\mathbf{u}^j(\cdot) = [M_1^j(\cdot) \quad M_2^j(\cdot) \quad M_3^j(\cdot) \quad F^j(\cdot)]^T, j = 1,2,3.$

Математические выражения для каждой компоненты вектора управления имеют вид

$$M_i = \begin{cases} M_i^-, \text{ если } \widetilde{M}_i < M_i^-, \\ M_i^+, \text{ если } \widetilde{M}_i > M_i^+, \\ \widetilde{M}_i \quad - \quad \text{иначе}, \end{cases} \qquad x_i^* = \begin{cases} x_i^-, \text{ если } \tilde{x}_i < x_i^-, \\ x_i^+, \text{ если } \tilde{x}_i > x_i^+, \\ \tilde{x}_i^* \quad - \quad \text{иначе}, \end{cases} \tag{4.76}$$

$$F = \begin{cases} F^-, \text{ если } \tilde{F} < F^-, \\ F^+, \text{ если } \tilde{F} > F^+, \\ \tilde{F} \quad - \quad \text{иначе}, \end{cases}$$

где

$$\widetilde{M}_1 = \left(q_4(x_4^* - x_4) + q_1(x_1^* - x_1) + (x_4^* - x_4)^3 + \sqrt[3]{q_1(x_1^* - x_1)}\right)^{-1} +$$

$$+\left(q_4(x_4^* - x_4) + q_1(x_1^* - x_1) + (x_4^* - x_4)^3 + \sqrt[3]{q_1(x_1^* - x_1)}\right)^{1/3} +$$

$$+\text{sgn}(x_6^* - x_6)\log(|q_6(x_6^* - x_6)| + 1) + q_2(x_2^* - x_2) + q_1(x_1^* - x_1) +$$

$$+\text{sgn}(x_4^* - x_4)\sqrt{|q_4(x_4^* - x_4)|} + (q_4(x_4^* - x_4))^3, \qquad (4.77)$$

$$\widetilde{M}_2 = \text{sgn}(\text{sgn}(A_1)\exp(|A_1| - 1)\sqrt{|\text{sgn}(A_1)\exp(|A_1| - 1)|}\,, \qquad (4.78)$$

$$\widetilde{M}_3 = \tanh(0.5B_1) - q_3(x_3^* - x_3) + \text{sgn}(x_5^* - x_5)\log(|q_5(x_5^* - x_5)| + 1) -$$

$$-x_3^* + x_3 + \sqrt[3]{B_1} + q_6(x_6^* - x_6) + q_2(x_2^* - x_2)\,, \qquad (4.79)$$

$$A_1 = C_1 + (q_3 + 1)(x_3^* - x_3)\,,$$

$$B_1 = (C_1 + q_3(x_3^* - x_3))^3 + q_6(x_6^* - x_6) + q_2(x_2^* - x_2)\,,$$

$$C_1 = \text{sgn}(x_5^* - x_5)\log(|q_5(x_5^* - x_5)| + 1),$$

$$\tilde{x}_1^* = \left(A_2 q_9(x_9^f - x_9)\cos(x_{11})exp(-q_{12})\right)\sqrt[3]{A_2}\log\left(\left|q_9(x_9^f - x_9)\cos(x_{11})\right|\right),$$

$$\tilde{x}_2^* = \sqrt[3]{\tilde{x}_1^*} + 2\arctan(C_2) - q_{10}^3 x_{10}^9 + D_2 - q_{11}x_{11}q_8^2(x_8^f - x_8)^2 - q_7(x_7^f - x_7) +$$

$$+ \arctan\left(A_2 q_9\left(x_9^f - x_9\right)\cos(x_{11})\exp(-q_{12})\right) + \arctan\left(-q_{10}x_{10}^3 + q_7\left(x_7^f - x_7\right)\right)$$

$$+q_8\left(x_8^f - x_8\right) + \text{sgn}\left(q_{10}x_{10}^3 + q_7\left(x_7^f - x_7\right)\right)\sqrt{\left|q_{10}x_{10}^3 + q_7\left(x_7^f - x_7\right)\right|} +$$

$$\tanh(-0.5x_{12}) + \text{sgn}(B_2)(\exp(|B_2|) - 1) +$$

$$\mu(A_2 q_9(x_9^f - x_9)\cos(x_{11})exp(-q_{12})),$$

$$\tilde{x}_3^* = \text{sgn}(\tilde{x}_1^*)\ln(|\tilde{x}_1^*| + 1) + \sqrt[3]{B_2} + \arctan(C_2) - q_{10}^3 x_{10}^9 - \tilde{x}_1^* +$$

$$(-q_{10}x_{10}^3 + q_7(x_7^f - x_7))^3\,,$$

$$\tilde{F} = \sin(\tilde{x}_3^*) + \text{sgn}(\tilde{x}_2^*)\exp(|\tilde{x}_2^*| - 1) + \text{sgn}(\tilde{x}_1^*)\ln(|\tilde{x}_1^*| + 1) + C_2^2 + \tanh(0.5B_2)$$

$$+ \left(A_2 q_9\left(x_9^f - x_9\right)\cos(x_{11})\exp(-q_{12})\right)^2 + \tanh(0.5E_2) +$$

$$(-q_{10}x_{10}^3 + q_7(x_7^f - x_7))^2\,, \qquad (4.80)$$

$$A_2 = q_{12}x_{12} + \sqrt[3]{q_{10}} + \arctan(q_9) + \cos(x_7^f - x_7)\,,$$

$$B_2 = \arctan(-q_{10}x_{10}^3 + q_7(x_7^f - x_7)) - q_7(x_7^f - x_7) - q_{11}x_{11}q_8^2(x_8^f - x_8)^2 +$$

$$q_8(x_8^f - x_8)\,,$$

$$C_2 = \text{sgn}(-q_{10}x_{10}^3 + q_7(x_7^f - x_7))(\exp(|-q_{10}x_{10}^3 + q_7(x_7^f - x_7)| - 1) + q_7^3 +$$
$$+ \text{sgn}(x_{10})\sqrt{|q_{10}x_{10}^3|} + \mu(q_7(x_7^f - x_7)),$$
$$D_2 = 2(\arctan(-q_{10}x_{10}^3 + q_7(x_7^f - x_7)) - q_{11}x_{11}q_8^2(x_8^f - x_8)^2 + q_8(x_8^f - x_8) - 1$$
$$+ \text{sgn}(-q_{11}x_{11}q_8^2(x_8^f - x_8)^2)\exp(-|-q_{11}x_{11}q_8^2(x_8^f - x_8)^2|) +$$
$$+\exp(q_{10}) + (-q_{10}x_{10}^3 + q_7(x_7^f - x_7))^3 - q_7(x_7^f - x_7)),$$
$$\mu(\alpha) = \begin{cases} \text{sgn}(\alpha), & \text{если } |\alpha| > 1 \\ \alpha \text{ - иначе} \end{cases}, \tanh(\alpha) = \frac{1-\exp(-2\alpha)}{1+\exp(-2\alpha)},$$

$q_1 = 12.224,\ q_2 = 14.197,\ q_3 = 13.661,\ q_4 = 4.361,\ q_5 = 9.989,\ q_6 = 4.114,$
$q_7 = 0.115, q_8 = 3.371, q_9 = 3.076, q_{10} = 0.144, q_{11} = 3.131, q_{12} = 4.515.$

Решение получено в виде матрицы сетевого оператора и декодировано к виду (4.76) – (4.80). Для стабилизации объекта управления данные функции подставляются в правые части модели (4.53) объекта управления.

На Рис. 4.23 и 4.24 представлены графики движения квадрокоптера из восьми начальных условий в нулевое терминальное состояние. Использованы следующие начальные условия:

$$x^{0,1} = [0\ 0\ 0\ 0\ 0\ 0\ -0.5\ -0.5\ -0.5\ 0\ 0\ 0]^T,$$
$$x^{0,2} = [0\ 0\ 0\ 0\ 0\ 0\ -0.5\ -0.5\ 0.5\ 0\ 0\ 0]^T,$$
$$x^{0,3} = [0\ 0\ 0\ 0\ 0\ 0\ -0.5\ 0.5\ -0.5\ 0\ 0\ 0]^T,$$
$$x^{0,4} = [0\ 0\ 0\ 0\ 0\ 0\ -0.5\ 0.5\ 0.5\ 0\ 0\ 0]^T,$$
$$x^{0,5} = [0\ 0\ 0\ 0\ 0\ 0\ 0.5\ -0.5\ -0.5\ 0\ 0\ 0]^T,$$
$$x^{0,6} = [0\ 0\ 0\ 0\ 0\ 0\ 0.5\ -0.5\ 0.5\ 0\ 0\ 0]^T,$$
$$x^{0,7} = [0\ 0\ 0\ 0\ 0\ 0\ 0.5\ 0.5\ -0.5\ 0\ 0\ 0]^T,$$
$$x^{0,8} = [0\ 0\ 0\ 0\ 0\ 0\ 0.5\ 0.5\ 0.5\ 0\ 0\ 0]^T.$$

Выбранные начальные условия представляют собой вершины куба в трехмерном пространстве, а графики на рис.4.23 и 4.24 показывают проекции пространственных траекторий на указанные плоскости.

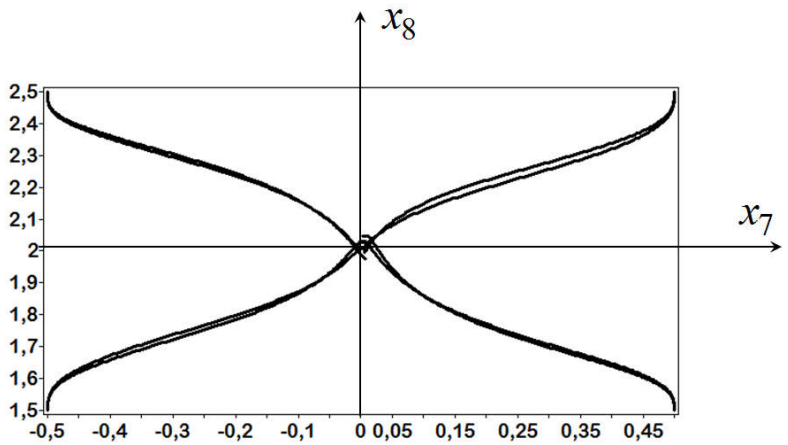

Рис. 4.41. Проекции траекторий движения квадрокоптера на плоскость $\{x_7, x_8\}$

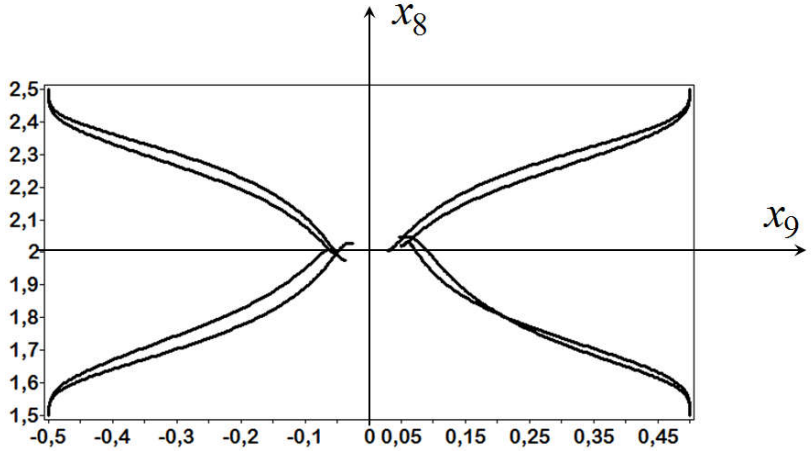

Рис. 4.42. Проекции траекторий движения квадрокоптера на плоскость $\{x_9, x_8\}$

На втором этапе задача оптимального управления решается как задача поиска оптимального положения точек стабилизации. Поскольку положения двух любых квадрокоптеров зависят от положения третьего квадрокоптера и параметра R_0 равностороннего треугольника, необходимо найти следующие оптимальные значения параметров:

$$(x_7^{1,*,1}, x_8^{1,*,1}, x_9^{1,*,1}, R_0^{*,1}, \tau^{*,1}), \ldots, (x_7^{1,*,K}, x_8^{1,*,K}, x_9^{1,*,K}, R_0^{*,K}, \tau^{*,K}), \qquad (4.81)$$

где K – количество точек стабилизации.

Положение других квадрокоптеров определяется по значениям найденных оптимальных параметров из уравнений:

$$x_7^{2,*,i} = x_7^{1,*,i} - 3R_0^{*,i}\cos(\tau^{*,i})/2 + \sqrt{3}R_0^{*,i}sin(\tau^{*,i})/2,$$
$$x_8^{2,*,i} = x_8^{1,*,i}, \quad (4.82)$$
$$x_9^{2,*,i} = x_9^{1,*,i} - 3R_0^{*,i}sin(\tau^{*,i})/2 - \sqrt{3}R_0^{*,i}\cos(\tau^{*,i})/2,$$
$$x_7^{3,*,i} = x_7^{1,*,i} - 3R_0^{*,i}\cos(\tau^{*,i})/2 - \sqrt{3}R_0^{*,i}sin(\tau^{*,i})/2,$$
$$x_8^{3,*,i} = x_8^{1,*,i}, \quad (4.83)$$
$$x_9^{3,*,i} = x_9^{1,*,i} - 3R_0^{*,i}sin(\tau^{*,i})/2 + \sqrt{3}R_0^{*,i}\cos(\tau^{*,i})/2,$$

где $i = 1, \dots, K$.

Для поиска оптимальных параметров (4.81) использовался эволюционный алгоритм оптимизации роя частиц (PSO). В качестве целевой функции использовался исходный функционал (4.75), который вычисляется при моделировании системы (4.53) с функциями управления (4.76)-(4.80) в правых частях.

Для искомых параметров (4.81) использовались следующие ограничения: $x_7^{1,*,i} \in (0{,}12)$, $x_8^{1,*,i} \in (0{,}8)$, $x_9^{1,*,i} \in (0{,}12)$, $R_0^{*,i} \in (0.8,\ 1.5)$, $\tau^{*,i} \in (-\pi/4,\ \pi/4)$. Искомые параметры (4.81) использовались группами по 5 параметров: первые три параметра $x_7^{1,*,i}$, $x_8^{1,*,i}$, $x_9^{1,*,i}$ указывали координаты точки стабилизации, остальные параметры $R_0^{*,i}$, $\tau^{*,i}$ определяли конфигурацию строя. Переключение между группами параметров осуществляется через заданный интервал 0.8 сек. Предельное время процесса управления $t^+ = 4.0$, поэтому число интервалов $K = 5$. Всего оптимизируемых параметров 25. Использованы следующие значения свободных параметров алгоритма PSO: количество возможных решений в начальном наборе 64, число эволюционных циклов 16384.

Для проведения вычислительного эксперимента были заданы следующие параметры.

Начальные условия

$$x^1(0) = [0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0.67\ \ 2\ \ 0\ \ 0\ \ 0\ \ 0]^T,$$
$$x^2(0) = [0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ -0.33\ \ 2\ \ -0.87\ \ 0\ \ 0\ \ 0]^T,$$
$$x^3(0) = [0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ -0.33\ \ 2\ \ -0.87\ \ 0\ \ 0\ \ 0]^T.$$

Терминальное положение груза

$$y^f = [10\ \ 1\ \ 10\ ]^T.$$

Длина тяги $d = 1.8$ м. Число статических ограничений $s = 4$. Параметры ограничений: $r_1 = 1,\ r_2 = 1,\ r_3 = 1,\ r_4 = 1,\ x_{1,1} = 2.5,\ x_{2,1} = 2.5,\ x_{1,2} = 7.5,\ x_{2,2} = 7.5,\ x_{1,3} = 2,\ x_{2,3} = 8,\ x_{1,4} = 8,\ x_{4,1} = 2.$ Весовые коэффициенты: $a_1 = 2.5,\ a_2 = 2.5,\ a_3 = 2.5.$ Другие параметры: $\varepsilon_p = 0.05, \varepsilon_r = 0.1, t^+ = 4.0, K = 5.$

В результате были получены следующие значения параметров: $x_7^{1,*,1} = -1.137,\ x_8^{1,*,1} = 2.257,\ x_9^{1,*,1} = 0.85,\ R_0^{*,1} = 1.919,\ \tau^{*,1} = -1.57,$ $x_7^{1,*,2} = 11.072,\ x_8^{1,*,2} = 4.597,\ x_9^{1,*,2} = 10.936,\ R_0^{*,2} = 0.842,\ \tau^{*,2} = -0.457,$ $x_7^{1,*,3} = 4.786,\ x_8^{1,*,3} = 3.303,\ x_9^{1,*,3} = 2.556,\ R_0^{*,3} = 0.535,\ \tau^{*,3} = 1.508,$ $x_7^{1,*,4} = 5.734, x_8^{1,*,4} = 4.295, x_9^{1,*,4} = -1.80, R_0^{*,4} = 1.416, \tau^{*,4} = -0.745, x_7^{1,*,5} = -2.00, x_8^{1,*,5} = 1.261, x_9^{1,*,5} = 5.957, R_0^{*,1} = 0.676, \tau^{*,1} = 1.305.$

На рис. 4.43 представлены проекции полученных траекторий квадрокоптера на горизонтальную плоскость $\{x_7, x_9\}$.

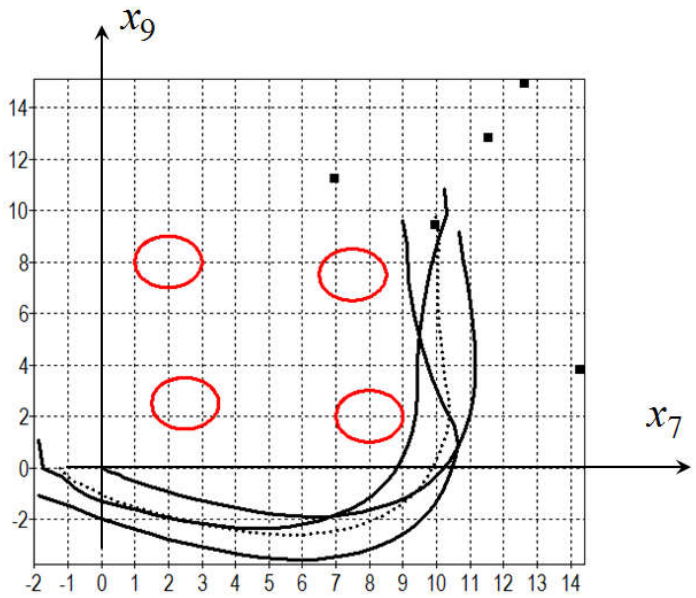

Рис. 4.43. Траектории движения квадрокоптеров и груза на плоскости $\{x_7, x_9\}$

Точечная кривая показывает траекторию движения груза. Красные круги – препятствия. Сплошные линии – траектории движения квадрокоптеров. Из

рисунка видно, что квадрокоптеры и груз не касаются препятствий. Груз достиг конечной точки без столкновений и нарушения фазовых ограничений.

Итак, в данном подразделе была рассмотрена еще одна задача управления. Особенностью данной задачи является еще более сложное описание модели объекта управления и особенно функционала качества управления, который явно не отвечает требованиям гладкости и унимодальности. Три квадрокоптера должны перемещать груз из исходного состояния в конечную точку без столкновений и с оптимальным значением критерия качества. Для решения задачи использован подход на основе принципа синтезированного оптимального управления. Согласно этому подходу, в первую очередь решается задача синтеза системы стабилизации квадрокоптеров. Для решения этой задачи используется автоматический метод машинного обучения на основе символьной регрессии. В задаче синтеза системы стабилизации математическая модель квадрокоптера разбита на две более простые модели - модели углового и пространственного движений. Поскольку пространственное управление квадрокоптером осуществляется путем наклона его на некоторый угол, синтез системы угловой стабилизации был решен в первую очередь. После этого была решена задача синтеза системы пространственной стабилизации квадрокоптера. Для решения задач синтеза использовался метод сетевого оператора, который позволил найти нелинейные функции управления по состоянию объекта, опираясь при поиске на минимизацию значения функционала качества. На втором этапе были найдены оптимальные положения точек стабилизации в пространстве состояний. При поиске этих точек в критерии качества учитывались и фазовые ограничения. Для поиска точек стабилизации использовался эволюционный алгоритм. Расчетный пример показал эффективность предложенного подхода в задаче со сложным видом объекта и функционала.

4.6. Сравнение эволюционных алгоритмов при расчете синтезированного оптимального управления

В предыдущих разделах были рассмотрены различные задачи управления, отличающиеся как самими робототехническими объектами, так и условиями функционирования. Применяемые в задачах функционалы не являются унимодальным на пространстве искомых параметров, особенно при наличии фазовых ограничений. Поэтому для решения задач использовались эволюционные алгоритмы. В данном разделе проведем экспериментальное сравнительное исследование различных эволюционных алгоритмов в задаче управления группой мобильных роботов с фазовыми ограничениями. Рассмотрим эволюционные алгоритмы, которые на сегодняшний день наиболее популярны: генетический алгоритм GA, алгоритм роя частиц PSO, алгоритм самоорганизующейся миграции SOMA, модифицированный SOMA и алгоритм серых волков GWO. Основные механизмы их работы представлены ранее в разделе 2.5.5.

Для проведения сравнительного вычислительного эксперимента рассмотрим задачу оптимального управления группой из четырех роботов, перемещающихся в едином ограниченном пространстве. Предполагаем, что каждый робот знает положение другого робота и согласует свое перемещение с учетом перемещений других роботов в группе.

Математическая модель движения роботов можно представить следующей системой дифференциальных уравнений:

$$\begin{aligned} \dot{x}_1^j &= 0.5(u_1^j + u_2^j)\cos(x_3^j), \\ \dot{x}_2^j &= 0.5(u_1^j + u_2^j)\sin(x_3^j), \\ \dot{x}_3^j &= 0.5(u_1^j - u_2^j), \end{aligned} \tag{4.84}$$

где $\mathbf{x}^j = \left[x_1^j \;\; x_2^j \;\; x_3^j\right]^T$ - вектор состояния робота j, $\mathbf{u}^j = \left[u_1^j \;\; u_2^j\right]^T$ - вектор управления роботом j, $j = 1, \dots, N = 4$.

Заданы ограничения на управление:

$$u_i^- \le u_i^j \le u_i^+, i = 1,2, j = 1, \dots, N, \quad (4.85)$$

где u_i^-, u_i^+ - заданные значения ограничений на управление, $u_i^- = -10, u_i^+ = 10$, $i = 1,2$.

Заданы начальные состояния роботов

$$\mathbf{x}^{1,0} = [0\ \ 0\ \ 0]^T, \mathbf{x}^{2,0} = [0\ \ 10\ \ 0]^T,$$
$$\mathbf{x}^{3,0} = [10\ \ 0\ \ 0]^T, \mathbf{x}^{4,0} = [10\ \ 10\ \ 0]^T. \quad (4.86)$$

Заданы целевые терминальные положения:

$$\mathbf{x}^{1,f} = [10\ \ 10\ \ 0]^T, \mathbf{x}^{2,f} = [10\ \ 0\ \ 0]^T,$$
$$\mathbf{x}^{3,f} = [0\ \ 10\ \ 0]^T, \mathbf{x}^{4,f} = [0\ 0\ 0]^T. \quad (4.87)$$

Предположим, что в пространстве движения роботов заданы фазовые ограничения:

$$r_k^2 - \left(x_1^j - x_{1,k}\right)^2 - \left(x_2^j - x_{2,k}\right)^2 \le 0, j = 1,2,3, \quad (4.88)$$

где $r_k = 1.5, k = 1,2,3,4, x_{1,1} = 2.5,\ x_{2,1} = 2.5,\ x_{1,2} = 7.5,\ x_{2,2} = 7.5,\ x_{1,3} = 2.5,$ $x_{2,3} = 7.5,\ x_{1,4} = 7.5,\ x_{2,4} = 2.5.$

Динамические фазовые ограничения определяются условием отсутствия столкновений между роботами в группе:

$$r_0^2 - \left(x_1^k - x_1^j\right)^2 - \left(x_2^k - x_2^j\right)^2 \le 0, \quad (4.89)$$

где $k = 1,2,3, j = k + 1, \dots,4, r_0 = 1.$

Задан критерий качества:

$$J = t_f = \max\{t_{1,f}, t_{2,f}, t_{3,f}, t_{4,f}\} \to \min, \quad (4.90)$$

где $t_{j,f}$ определяется по формуле (12) при $t^+ = 2{,}8$ сек, $\varepsilon = 0{,}01$.

Используем принцип синтезированного оптимального управления для решения поставленной задачи (4.84) - (4.90).

Согласно принципу, на первом этапе решается задача синтеза системы стабилизации. Поскольку роботы в группе в рассматриваемой задаче одинаковые, то задачу синтеза решаем для одного робота. Для решения используем метод вариационного декартова генетического программирования,

представленного в разделе 3.7.5, который является универсальным методом символьной регрессии для решения задач подобного класса [220].

При решении задачи синтеза фазовые ограничения (4.88) и (4.89) не учитывались. Было задано двадцать семь начальных условий:

$$\mathbf{x}^{1,0,1} = \left[-5 \;\; -5 \;\; -\frac{\pi}{2}\right]^T, \mathbf{x}^{1,0,2} = [-5 \;\; -5 \;\; 0]^T, \mathbf{x}^{1,0,3} = \left[-5 \;\; 5 \;\; \frac{\pi}{2}\right]^T,$$

$$\mathbf{x}^{1,0,4} = \left[-5 \;\; 0 \;\; -\frac{\pi}{2}\right]^T, \mathbf{x}^{1,0,5} = [-5 \;\; 0 \;\; 0]^T, \; \mathbf{x}^{1,0,6} = \left[-5 \;\; 0 \;\; \frac{\pi}{2}\right]^T,$$

$$\mathbf{x}^{1,0,7} = \left[-5 \;\; 5 \;\; -\frac{\pi}{2}\right]^T, \mathbf{x}^{1,0,8} = [-5 \;\; 5 \;\; 0]^T, \mathbf{x}^{1,0,9} = \left[-5 \;\; 5 \;\; \frac{\pi}{2}\right]^T,$$

$$\mathbf{x}^{1,0,10} = \left[0 \;\; -5 \;\; -\frac{\pi}{2}\right]^T, \mathbf{x}^{1,0,11} = [0 \;\; -5 \;\; 0]^T, \mathbf{x}^{1,0,12} = \left[0 \;\; 5 \;\; \frac{\pi}{2}\right]^T,$$

$$\mathbf{x}^{1,0,13} = \left[0 \;\; 0 \;\; -\frac{\pi}{2}\right]^T, \mathbf{x}^{1,0,14} = [0 \;\; 0 \;\; 0]^T, \; \mathbf{x}^{1,0,15} = \left[0 \;\; 0 \;\; \frac{\pi}{2}\right]^T,$$

$$\mathbf{x}^{1,0,16} = \left[0 \;\; 5 \;\; -\frac{\pi}{2}\right]^T, \mathbf{x}^{1,0,17} = [0 \;\; 5 \;\; 0]^T, \mathbf{x}^{1,0,18} = \left[0 \;\; 5 \;\; \frac{\pi}{2}\right]^T,$$

$$\mathbf{x}^{1,0,19} = \left[5 \;\; -5 \;\; -\frac{\pi}{2}\right]^T, \mathbf{x}^{1,0,20} = [5 \;\; -5 \;\; 0]^T, \mathbf{x}^{1,0,21} = \left[5 \;\; 5 \;\; \frac{\pi}{2}\right]^T,$$

$$\mathbf{x}^{1,0,22} = \left[5 \;\; 0 \;\; -\frac{\pi}{2}\right]^T, \mathbf{x}^{1,0,23} = [5 \;\; 0 \;\; 0]^T, \mathbf{x}^{1,0,24} = \left[5 \;\; 0 \;\; \frac{\pi}{2}\right]^T,$$

$$\mathbf{x}^{1,0,25} = \left[5 \;\; 5 \;\; -\frac{\pi}{2}\right]^T, \mathbf{x}^{1,0,26} = [5 \;\; 5 \;\; 0]^T, \mathbf{x}^{1,0,27} = \left[5 \;\; 5 \;\; \frac{\pi}{2}\right]^T.$$

Система стабилизировалась в точку $\mathbf{x}^{1,f,1} = [0 \;\; 0 \;\; 0]^T$.

Критерий качества без учета фазовых ограничений имел следующий вид:

$$J_2 = \sum_{k=1}^{27} t_{\{1,f,k\}} \to \min. \tag{4.81}$$

В результате было получено следующее математическое выражение для функции управления:

$$u_i = \begin{cases} u_i^-, \text{если } \tilde{u}_i < u_i^-, \\ u_i^+, \text{если } \tilde{u}_i > u_i^+, \\ \tilde{u}_i - \text{иначе}, \end{cases} \quad i = 1,2, \tag{4.82}$$

где

$$\tilde{u}_1 = A + B + \operatorname{sgn}((A)\exp(|A|) - 1),$$

$$\tilde{u}_2 = B - A - \operatorname{sgn}((A)\exp(|A|) - 1),$$

$$A = q_1(x_3^* - x_3) + \operatorname{sgn}\big((x_1^* - x_1)(x_2^* - x_2)\big) \times \sqrt{|(x_1^* - x_1)(x_2^* - x_2)|},$$

$$B = 2(x_1^* - x_1) + \mathrm{sgn}(x_1^* - x_1)q_2,$$
$$q_1 = 3.109,\ q_2 = 3.629.$$

На втором этапе необходимо было найти оптимальное положение точек равновесия $\mathbf{x}^{j,*,k}$ для всех четырёх роботов. Для каждого робота искали по три точки $k = 1,2,3$. При решении задачи оптимального управления использовался следующий критерий качества:

$$J_3 = t_f + a_1 \sum_{k=1}^{3} \sum_{j=k+1}^{4} \int_0^{t_f} \vartheta\left(\chi\left(\mathbf{x}^k, \mathbf{x}^j\right)\right) dt + \sqrt{0.25 \sum_{i=1}^{4} \sum_{j=1}^{4} \left(x_j^{i,f} - x_1^i\right)^2}, \quad (4.83)$$

где

$$\vartheta(A) = \begin{cases} 1, \text{если A} > 0, \\ 0 - \text{иначе.} \end{cases}$$

Использовались описанные в предыдущем разделе алгоритмы: генетический алгоритм (GA), алгоритм серых волков (GWO), алгоритм роя частиц (PSO), алгоритм самоорганизующейся миграции (SOMA) и модифицированный алгоритм самоорганизующейся миграции (ModSOMA). Каждая точка имеет три координаты. Поэтому алгоритмы искали $p = 36$ параметров.

На параметры были заданы следующие ограничения:

$$q^-_{1+(j-1)3} = -2,\ q^-_{2+(j-1)3} = -2,\ q^-_{3+(j-1)3} = -1.57,$$
$$q^-_{1+(j-1)3} = 12,\ q^-_{2+(j-1)3} = 12,\ q^-_{3+(j-1)3} = 1.57,$$

где $j = 1,2,3$.

Параметры алгоритмов были заданы следующие.

GA: $H = 4096$, число поколений $P = 256$, количество операций скрещивания в одном поколении $R = 256$, $P_\mu = 0.75$, $D = 32$;

SOMA: $H = 128$, число поколений $P = 32$, $P_l = 8$, $s_1 = 0.22$, $P_{rt} = 0.33$;

ModSOMA: $H = 128$, количество поколений $P = 32$, $P_l = 8$, $s_1 = 0.22$, $P_{rt} = 0.33$, $w = 7$, $a_1 = 0.3$, $a_2 = 0{,}5$;

GWO: $H = 128$, количество поколений $P = 1024$, $k_w = 3$;

PSO: $H = 128$, количество поколений $P = 1024$, количество модификаций вектора параметров в одно поколение $R = 128$, $\alpha = 0.7298$, $\beta = 0.85$, $\gamma = 0.15$, $\sigma = 1$.

Каждый алгоритм запускался десять раз. Для каждого алгоритма определялись среднее значение функционала, лучшее значение и среднеквадратическое значение на всех прогонах. Результаты экспериментов представлены в Таблице 4.6. Также в последней строке таблицы представлены количество вычислений функционала на всех прогонах.

Таблица 4.5. Результаты вычислительного эксперимента

	GA	GWO	PSO	SOMA	modSOMA
Среднее знач.	3,35	2,90	2,94	3,28	2,91
СКО	0,526	0,051	0,274	0,384	0,063
Лучшее знач.	2,93	2,81	2,77	2,97	2,79
Кол-во расч. функ.	1326133	1296650	1312011	1475851	1475851

Алгоритмы PSO, ModSOMA и GWO показали примерно одинаковые результаты. Лучшее решение нашел PSO. GA и SOMA показали между собой одинаковые результаты, но они оказались немного хуже остальных. GWO показал лучшую стабильность работы по среднему значению и по среднеквадратичному отклонению.

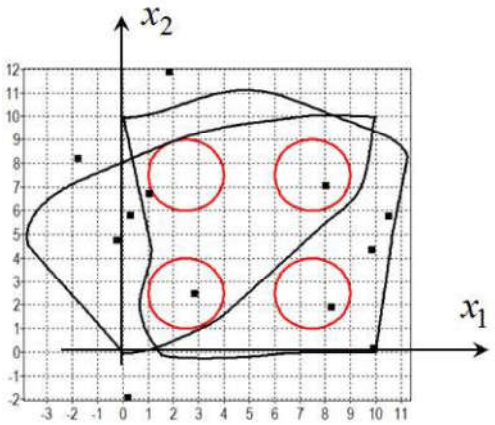

Рис.4.44. Траектории роботов, полученные с помощью GA

На рис. 4.44 – 4.48 представлены траектории движения роботов. Как видно на представленных графиков, траектории, полученные всеми исследуемыми эволюционными алгоритмами, достигают целевых терминальных положений без столкновений между роботами и не нарушая фазовых ограничений.

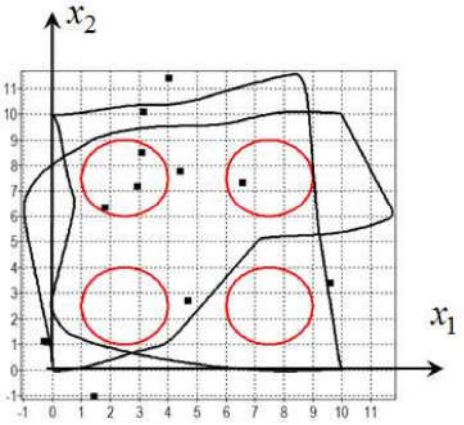

Рис.4.45. Траектории роботов, полученные с помощью GWO

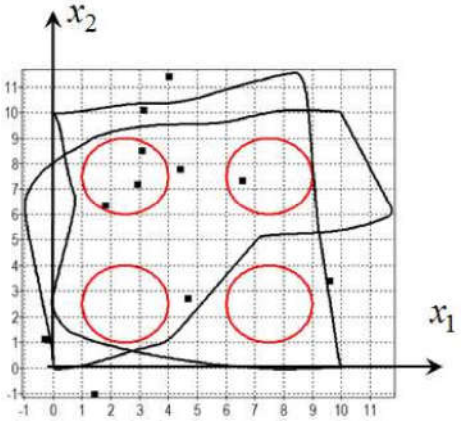

Рис.4.46. Траектории роботов, полученные алгоритмом SOMA

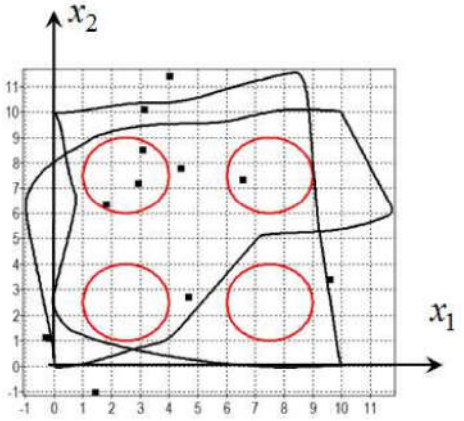

Рис.4.47. Траектории роботов, полученные алгоритмом modSOMA

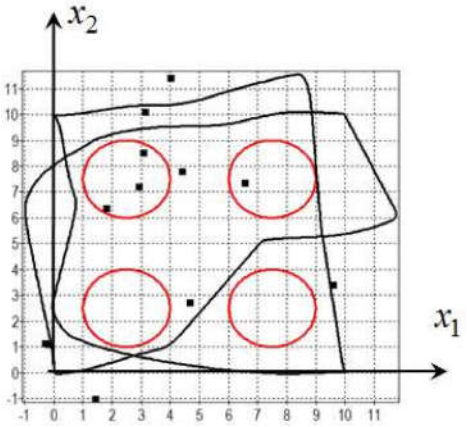

Рис.4.48. Оптимальные траектории роботов, полученные с помощью PSO

Для решения этой задачи также был применен один из наиболее популярных сегодня градиентных алгоритмов – алгоритм сстохастического градиента ADAM [236], который хорошо зарекомендовал себя при настройке параметров нейронной сети.

Для проведения вычислительного эксперимента были заданы следующие значения параметров этого алгоритма: $\rho_{1,i} = 0.9$, $\rho_{2,i} = 0.999$, $i = 1, \dots, 36$, $\delta = 0.01$, $\varepsilon = 0.01$. ADAM не смог найти приемлемого решения поставленной задачи. Лучший результат, полученный ADAM после более чем 1000000 запусков, составил 6,11.

При сравнении все алгоритмы должны были вычислить целевой функционал примерно одинаковое количество раз. Критериями сравнения служили значение функционала для наилучшего найденного решения, среднее значение функционала по всем тестам и стандартное отклонение функционала. Как показали представленные результаты, алгоритм роя частиц нашел наилучшее решение. Алгоритм серых волков и модифицированный алгоритм самоорганизующейся миграции также показали хорошие результаты. Эти алгоритмы учитывают большее количество факторов при выполнении операции эволюции, чем другие. При этом все рассмотренные эволюционные алгоритмы справились с решением поставленной задачи оптимального управления с фазовыми ограничениями, а используемый для сравнения градиентный алгоритм ADAM не смог решить поставленную задачу за разумное время.

4.7. Выводы

В экспериментальной части диссертации были решены различные задачи управления робототехническими системами, такие как управление мобильным роботом в условиях фазовых ограничений, группой мобильных роботов, где в постановку задачи добавляются динамические фазовые ограничения, а также задача управления для робота с необычным типом движения и управления, для задачи управления движением квадрокоптера и группы квадрокоптеров в задаче совместного взаимодействия. Представленные решения задач демонстрируют эффективность предложенного подхода к автоматической разработке систем управления на основе принципа синтезированного оптимального управления.

Преимущества такого подхода состоят в том, что все этапы разработки реализуются автоматически на ЭВМ, исходя из сформулированной математической постановки задачи. Подход является универсальным и не ограничивается определенными типами моделей объектов управления или функционалов качества управления. При этом за счет численного синтеза системы стабилизации на первом этапе удается нивелировать некоторые неопределенности моделей объектов управления и начальных условий. Результаты вычислительных экспериментов с разработанными системами управления, представленные в таблицах, показали, что для систем управления, полученных на основе принципа синтезированного оптимального управления, возмущения модели и начальных условий влияют на ухудшение значения функционала в пределах 25%, а для систем оптимального управления, полученных на основе прямого подхода, для того уже уровня возмущений превышает 100%, при этом значение среднеквадратичного отклонения для синтезированного оптимального управления меньше в 3 раза, чем для прямого.

Заключение

В работе решена важная научно-техническая проблема разработки автоматизированных подходов к созданию систем управления робототехническими объектами, оптимальных с точки зрения заданного критерия качества и реализуемых для практических объектов.

Представлены формальные постановки задач в терминах функционала качества, решаемые в робототехнике. Проведен анализ существующих методов решения задач оптимального управления робототехническими системами и существующих подходов к их практической реализации. Показаны трудности применения разомкнутых типов управлений, получаемых в результате решения задач оптимального управления, и необходимость учета реального состояния объекта управления, позволяющая отрабатывать неопределенности, как начальных условий, так и модели объекта управления, используемой в расчетах оптимальных режимов управления. Показана необходимость и актуальность разработки современных универсальных численных подходов, позволяющих автоматизировать с помощью ЭВМ решение строгих математических задач управления.

Сформулирован принцип синтезированного оптимального управления, обеспечивающий получение практически реализуемого решения задачи оптимального управления. Представлена математическая формулировка предложенного подхода на основе принципа синтезированного оптимального управления. Согласно введенному принципу, решение задачи оптимального управления производится для объекта стабилизированного относительно точки равновесия в пространстве состояний.

Разработанный принцип синтезированного оптимального управления отвечает современным требованиям цифровой трансформации, автоматизирует процесс создания систем управления за счет внедрения универсальных технологий машинного обучения при использовании классических формулировок задач управления.

Разработан двухэтапный подход реализации принципа синтезированного оптимального управления. Первоначально решается задача синтеза управления, чтобы обеспечить устойчивость объекта относительно точки в пространстве состояний, а затем решается задача оптимального управления. Оптимальное управление реализуется за счет оптимального изменения положения устойчивой точки равновесия. Представлены основные преимущества принципа синтезированного оптимального управления. Экспериментально для различных робототехнических объектов продемонстрирована универсальность предлагаемого подхода и его применимость к различным задачам оптимального управления и различным объектам и функционалам.

Введено и формализовано обоснование применения принципа синтезированного управления для получения решения задачи оптимального управления, обладающего свойством реализуемости. Применение принципа оптимального синтезированного управления обеспечивает в каждый момент времени существование устойчивой точки равновесия. В окрестности точки равновесия фазовые траектории модели объекта управления сжимаются, что определяет реализуемость системы. Сжимаемость траекторий обеспечивает уменьшение расхождений между реальным объектом и его моделью при приближении к точке устойчивости.

Разработаны новые численные методы реализации этапов синтезированного оптимального управления. На первом этапе решается задача синтеза системы стабилизации объекта управления с целью обеспечения его устойчивости относительно точки равновесия в пространстве состояний. Для решения этой задачи используется машинное обучение управления методами символьной регрессии. На втором этапе решается задача оптимального размещения точек равновесия. Для решения этой задачи используются эволюционные алгоритмы. Разработанные численные методы расчета синтезированного оптимального управления являются универсальными, не зависит от типа модели объекта управления и целевого функционала, что обеспечивает автоматизацию процесса построения реализуемой системы

управления, причем задача оптимального размещения точек равновесия может решаться в режиме реального времени. Создание системы управления, согласно разработанному подходу, происходит автоматически средствами машинного обучения, что позволяет уменьшить объем ручного проектирования при разработке законов управления робототехническими системами.

Разработаны методы машинного обучения управления на основе символьной регрессии для реализации этапа синтеза системы стабилизации в рамках принципа синтезированного оптимального управления. Синтез системы стабилизации на первом этапе производится численно с помощью современных методов машинного обучения управления на основе символьной регрессии, которые позволяют находить и структуру, и параметры математического выражения функции управления, в отличие от известных подходов, когда структура функции задана и определяются только параметры. Разработанные методы символьной регрессии реализуют поиск математических выражений функции управления в виде определенного кода, в зависимости от метода, с помощью специального генетического алгоритма. Найденное математическое выражение, описывающее систему стабилизации, после подстановки в правые части системы дифференциальных уравнений модели объекта управления обеспечивает наличие в пространстве состояний устойчивой точки равновесия. Разработаны новые вариационные методы машинного обучения на основе символьной регрессии для синтеза системы стабилизации робототехническими объектами, предложены уникальные типы малых вариаций и способы их кодирования. Разработаны программные комплексы их реализации.

Разработаны методы реализации этапа оптимального расположения точек равновесия в рамках принципа синтезированного оптимального управления. Согласно принципу синтезированного оптимального управления, синтезированная функция управления имеет набор параметров, влияющих на положение точки равновесия, которые на втором этапе оптимизируются по критерию задачи оптимального управления на основе современных эволюционных и популяционных подходов. Представленные подходы

позволяют гарантированно получать решения, близкие к оптимальным, даже в задачах со сложным пространством поиска при отсутствии условий унимодальности и выпуклости функционала. Причем задача оптимального размещения точек равновесия может решаться непосредственно на борту объекта.

Введено формальное представление машинного обучения управления как поиска неизвестной функции управления, предложена методика машинного обоснования существования определенных свойств математической модели, в том числе машинно интерпретируемого свойства устойчивости точки в пространстве состояний. Для организации вычислений на пространстве элементарных функций и избегания вычислительных ошибок, таких как переполнение разрядной сетки, в работе вводится новое пространство машинно реализуемых функций.

Разработаны программные комплексы расчета синтезированного оптимального управления.

Решены различные задачи управления робототехническими системами, такие как управление мобильным роботом в условиях фазовых ограничений, группой мобильных роботов, где в постановку задачи добавляются динамические фазовые ограничения, задача управления для мобильного робота с меканум-колесами, задача управления движением квадрокоптера и группы квадрокоптеров в задаче совместного взаимодействия. Представленные решения задач демонстрируют эффективность предложенного подхода к автоматической разработке систем управления на основе принципа синтезированного оптимального управления.

Преимущества такого подхода состоят в том, что все этапы разработки реализуются автоматически на ЭВМ, исходя из сформулированной математической постановки задачи. Подход является универсальным и не ограничивается определенными типами моделей объектов управления или функционалов качества управления. При этом за счет численного синтеза системы стабилизации на первом этапе удается нивелировать некоторые

неопределенности моделей объектов управления и начальных условий. Вычислительные эксперименты с разработанными системами управления показали, что для систем управления, полученных на основе принципа синтезированного оптимального управления, возмущения модели и начальных условий влияют на ухудшение значения функционала в пределах 25%, а для систем оптимального управления, полученных на основе прямого подхода, для того уже уровня возмущений превышает 100%, при этом значение среднеквадратичного отклонения для синтезированного оптимального управления меньше в 3 раза, чем для прямого.

Полученные результаты моделирования движения различных робототехнических объектов с полученными системами управления демонстрируют эффективность предлагаемого принципа синтезированного оптимального управления в условиях присутствия неопределенностей модели объектов управления и начальных состояний, а также позволяют автоматизировать процесс разработки реализуемых систем управления, оптимальных с точки зрения заданных критериев, исходя из формальных математических постановок задач управления робототехническими объектами.

Литература

1. Понтрягин Л.С. Оптимальные процессы регулирования // Успехи матем. наук. – 1959. – Т. 14. – Вып. 1. – С. 3 - 20.
2. Dorf R.C., Bishop R.H. Modern Control Systems, Twelfth Edition. – Prentice Hall, 2011. – 1082 p.
3. Sharp N., Soliman Y., Crane K. Navigating intrinsic triangulations. // ACM Transactions on Graphics (TOG). – 2019. – 38. – pp. 1 - 16.
4. Li Z., Huang J. Study on the use of QR codes as landmarks for indoor positioning: Preliminary results // In Proceedings of the 2018 IEEE/ION Position, Location and Navigation Symposium (PLANS). – Monterey, CA, USA, 23-26 April, 2018. – pp. 1270-1276.
5. Novoselov S., Sychova O., Tesliuk S. Development of the Method Local Navigation of Mobile Robot a Based on the Tags with QR Code and Wireless Sensor Network // In Proceedings of the 2019 IEEE XVth International Conference on the Perspective Technologies and Methods in MEMS Design (MEMSTECH). – Polyana, Ukraine, 22-26 May, 2019. – pp. 46-51.
6. Zhou C., Liu X. The Study of Applying the AGV Navigation System Based on Two Dimensional Bar Code // In Proceedings of the 2016 International Conference on Industrial Informatics - Computing Technology, Intelligent Technology, Industrial Information Integration (ICIICII). – Wuhan, China, 3-4 December, 2016. – pp. 206-209.
7. Sani M.F., Karimian G. Automatic navigation and landing of an indoor AR drone quadrotor using ArUco marker and inertial sensors // In Proceedings of the 2017 International Conference on Computer and Drone Applications (IConDA). – Kuching, Malaysia, 9-11 November 2017. – pp. 102-107.
8. Marut A., Wojtowicz K., Falkowski K. ArUco markers pose estimation in UAV landing aid system // In Proceedings of the 2019 IEEE 5th International Workshop on Metrology for AeroSpace (MetroAeroSpace). – Turin, Italy, 19-21 June 2019. – pp. 261-266.

9. Tian W., Chen D., Yang Z., Yin H. The application of navigation technology for the medical assistive devices based on Aruco recognition technology. // In Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). – Las Vegas, NV, USA, 24 October 2020 - 24 January 2021. – pp. 2894-2899.
10. Гэн К.К., Чулин Н.А. Алгоритм навигации беспилотного летательного аппарата на основе улучшенного алгоритма одновременной локализации и картографирования с адаптивным локальным диапазоном наблюдения // Вестник МГТУ им. Н.Э. Баумана. Сер. Приборостроение. – 2017. – № 3. – С. 76–94.
11. Won D.H., Chun S., Sung S., Kang T., Lee Y.J. Improving mobile robot navigation performance using vision based SLAM and distributed filters. // Proceedings of the 2008 International Conference on Control, Automation and Systems. – Seoul, Republic of Korea, 14-17 October, 2008. – pp. 186-191.
12. Cheeseman P., Smith R., Self M. A stochastic map for uncertain spatial relationships // In Proceedings of the 4th International Symposium on Robotics Research. – Santa Cruz, CA, USA, 9-14 August, 1987. – pp. 467–474.
13. Lu F., Milios E. Globally Consistent Range Scan Alignment for Environment Mapping // Autonomous Robots. – 1997. – Volume 4. – pp. 333–349.
14. Biswas J.,Veloso M. Depth camera based indoor mobile robot localization and navigation // Proceedings of the IEEE International Conference on Robotics and Automation. – St. Paul, MN, USA, 14-18 May, 2012. – pp. 1697–1702.
15. Tu Y., Huang Z., Zhang X., Yu W., Xu Y., Chen B. The Mobile Robot SLAM Based on Depth and Visual Sensing in Structured Environment. // Robot Intelligence Technology and Applications. 3. Advances in Intelligent Systems and Computing; Kim, J.H., Yang, W., Jo, J., Sincak, P., Myung, H., Eds. – Springer: Cham, Switzerland. – 2015. – Volume 345. – pp. 343–357.
16. Kuutti S., Fallah S., Katsaros K., Dianati M., Mccullough F., Mouzakitis A. A survey of the state-of-the-art localization techniques and their potentials for

autonomous vehicle applications // IEEE Internet Things J. – Apr. 2018. – Vol. 5. – No. 2. – pp. 829-846.

17. Gatesichapakorn S., Takamatsu J., Ruchanurucks M. ROS based autonomous mobile robot navigation using 2d LiDAR and RGB-d camera // 2019 First International Symposium on Instrumentation Control Artificial Intelligence and Robotics (ICA-SYMP). – Jan. 2019. – pp. 151-154.
18. Kumar D., Muhammad N. A Survey on Localization for Autonomous Vehicles // IEEE Access. – 2023. – vol. 11. – pp. 115865-115883. doi: 10.1109/ACCESS.2023.3326069.
19. Шмалько Е.Ю., Прокопьев И.В., Дивеев А.И. Поддержка средств автономной навигации мобильного робота с помощью внутренней модели на нейронной сети // International Journal of Open Information Technologies. – 2023. – Т. 11. – № 2. – С. 25-31.
20. Bresson G., Alsayed Z., Yu L., Glaser S. Simultaneous Localization and Mapping: A Survey of Current Trends in Autonomous Driving // IEEE Transactions on Intelligent Vehicles. – 2017. – Vol. 2, No. 3. – pp. 194-220.
21. Egerstedt M. Motion Planning and Control of Mobile Robots. – Ph.D. Thesis. – Royal Institute of Technology, Stockholm, Sweden, 2000. – 180 с.
22. Понтрягин Л.С., Болтянский В.Г., Гамкрелидзе Р.В., Мищенко Е.Ф. Математическая теория оптимальных процессов. – М., Наука, 1969. – 384 с.
23. Милютин А.А., Дмитрук А.В., Осмоловский Н.П. Принцип максимума в оптимальном управлении. – Москва, Воробьевы Горы. Издательство ЦПИ при механико-математическом факультете МГУ, 2004. – 168 с.
24. Петров Ю. П. Вариационные методы теории оптимального управления. Изд. 2-е. – Л.: «Энергия», 1977. – 280 с.
25. Эльсгольц Л.Э. Вариационное исчисление. – М.: Гостехиздат, 1958. – 163 с.
26. Гельфанд И.М., Фомин С.В. Вариационное исчисление. – М.-Л.: Физматгиз, 1961. – 228 с.

27. Янг Л. Лекции по вариационному исчислению и теории оптимального управления. – М.: Мир, 1974. – 488 с.
28. Морозов С.Ф. Разрывные задачи вариационного исчисления: Учеб. пособие / Нижегород. гос. ун-т им. Н. И. Лобачевского. – Н. Новгород: ННГУ, 1991. – 79 с.
29. Кузнецов Ю.А., Семенов А.В. Избранные главы вариационного исчисления. Электронное учебно-методическое пособие. – Нижний Новгород: Нижегородский госуниверситет, 2012. – 69 с.
30. Блисс Г.А. Лекции по вариационному исчислению / Пер. с англ. Ю.К. Солнцева. Под ред. Л. Э. Эльсгольца. – Москва: Изд-во иностр. лит., 1950. – 348 с.
31. Лаврентьев М.А., Люстерник Л.А. Курс вариационного исчисления. – Москва; Ленинград: ГОНТИ, Ред. техн.-теоретич. лит-ры., 1938. – 192 с.
32. Зейферт Г., Трельфалль В. Вариационное исчисление в целом. – Ижевск: Издательский дом «Удмуртский университет», 2000. – 160 с.
33. Романко В.К. Курс дифференциальных уравнений и вариационного исчисления. – 2-е изд. – М.: Лаборатория Базовых Знаний, 2001. – 344 с.
34. Цлаф Л.Я. Вариационное исчисление и интегральные уравнения: 3-е изд., стереотипное. – СПб.: Лань, 2005. – 192 с.
35. Bertsekas D.P. Nonlinear Programming. 2nd edition. – Athena Scientific, Belmont, MA, 1999. – 791 с.
36. Федоренко Р.П. Приближенное решение задач оптимального управления – М.: Наука, 1978. – 488 с.
37. Болтянский В.Г. Математические методы оптимального управления. 2-е изд., перераб. и доп. – Москва: Наука, 1969. – 408 с.
38. Базара, Шетти. Нелинейное программирование. Теория и алгоритмы. / М. Базара, К. Шетти; пер. с англ. Т.Д. Березневой, В.А. Березнева. – Москва: Мир, 1982. – 583 с.

39. Аттетков А.В. Методы оптимизации: учебное пособие для студентов высших учебных заведений / А.В. Аттетков, В.С. Зарубин, А.Н. Канатников. – Москва : РИОР : ИНФРА-М, 2011. – 270 с.
40. Поляк Б.Т. Градиентные методы минимизации функционалов // Журнал вычислительной математики и математической физики. – 1963. – Т. 3. – №4. – С. 643-653.
41. Nesterov Y. Gradient methods for minimizing composite functions // Mathematical Programming. – 2013. – Volume 140. – pp. 125-161.
42. Гилл Ф., Мюррей У., Райт М. Практическая оптимизация. / Перевод с англ. В.Ю. Лебедева. – Москва: Мир, 1985. – 509 с.
43. Стронгин Р.Г. Численные методы в многоэкстремальных задачах:(Информационно-статистические алгоритмы). – М.: Наука. Гл. ред. физ.-мат. лит., 1978. – 240 с.
44. Horst R., Tuy H. Global optimization: Deterministic approaches. 3rd revised and enlarged ed. – Springer, 1996. – 748 с.
45. Граничин О. Н., Поляк Б. Т. Рандомизированные алгоритмы оптимизации и оценивания при почти произвольных помехах. – М.: Наука, 2003. – 291 с.
46. Evtushenko Yu.G., Posypkin M.A. A deterministic algorithm for global multi-objective optimization. // Optimization Methods and Software. – 2014. – 29:5. – pp. 1005-1019.
47. Евтушенко Ю.Г. Численный метод поиска глобального экстремума функций (перебор на неравномерной сетке) // Ж. вычисл. матем. и матем. физ. – 1971. – 11:6. – С. 1390-1403.
48. Glover F., Kochenberger G. A. (eds.). Handbook of metaheuristics. / International Series in Operations Research & Management Science. Vol 57. – Springer, Boston, MA, 2003. – 557 с.
49. Карпенко А.П. Современные алгоритмы поисковой оптимизации. Алгоритмы, вдохновленные природой. – М.: Издательство МГТУ им. Н.Э. Баумана, 2014. – 448 с.

50. Blum C., Aguilera M.J., Roli A., Sampels M. Hybrid Metaheuristics. An Emerging Approach to Optimization. / Studies in Computational Intelligence (SCI, volume 114). – Springer Velag, Berlin, Germany, 2008. – 289 p.
51. Michalewicz Z., Krawczyk J., Kazemi M., Janikow C.Z. Genetic algorithms and optimal control problems // Proceedings of the IEEE Conference on Decision and Control. – 1991. – 3. – pp. 1664 – 1666.
52. Дивеев А.И., Константинов С.В. Исследование практической сходимости эволюционных алгоритмов оптимального программного управления колесным роботом // Известия Российской академии наук. Теория и системы управления. – 2018. – № 4. – С. 75-98.
53. Athans M., Falb P.L. Optimal Control: An Introduction to the Theory and Its Applications. – McGraw Hill, New York, 1966. – 879 p.
54. Arutyunov A., Karamzin D., Pereira F.L. Optimal Impulsive Control. The Extension Approach. / Lecture Notes in Control and Information Sciences. – Springer, Cham, 2019. – 477 p.
55. Арутюнов А.В. К теории принципа максимума в задачах оптимального управления с фазовыми ограничениями. // Докл. АН СССР. – 1989. – Т. 304. – № 1. – С. 11-14.
56. Гамкрелидзе Р.В. Оптимальные процессы управления при ограниченных фазовых координатах. // Изв. АН СССР. – 1960. – Т. 24. – № 3. – С. 315–356.
57. Дубовицкий А.Я., Милютин А.А. Задачи на экстремум при наличии ограничений. // Докл. АН СССР. – 1963. – Т. 149. – № 4. – С. 759–762.
58. Арутюнов А.В., Тынянский Н.Т. О принципе максимума в задаче с фазовыми ограничениями. // Изв. АН СССР. Сер. техн. Кибернетика. – 1984. – № 4. – С. 60–68.
59. Bellman R.E. Dynamic Programming. – Princeton, NJ, USA: Princeton Univ. Press, 1957. – 342 p.

60. Моисеев Н.Н. Методы динамического программирования в теории оптимальных управлений. // Ж. вычисл. матем. и матем. физ. – 1964. – Т. 4. – № 3. – С. 485-494.
61. Bellman R.E, Kalaba R.E. Dynamic Programming and Modern Control Theory. – New York, London, Academic Press, 1966. – 112 p.
62. Bellman R.E., Dreyfus S.E. Applied Dynamic Programming. – Princeton University Press. Princet, 1962. – 390 p.
63. Габасов Р., Кириллова Ф. М. Основы динамического программирования – Минск: Изд-во БелГУ, 1975. – 262 с.
64. Bertsekas D.P. Dynamic Programming and Optimal Control. – Athena Scientific, USA, 2005. – 445 p.
65. Летов А.М. Аналитическое конструирование регуляторов. // Автоматика и телемеханика. – 1960. – Т. 21. – Вып. 4. – С.436–441.
66. Летов А.М. Математическая теория процессов управления. – М.: Наука, 1981. – 256 с.
67. Калман Р.Е. Когда линейная система является оптимальной? // Труды американского общества инженеров механиков. – Серия Д, №1: Мир. – 1964. – С. 69-84.
68. Красовский А.А. Системы автоматического управления полетом и их аналитическое конструирование. – М.: Наука, 1973. – 560 с.
69. Афанасьев В.Н. Оптимальные системы управления. Аналитическое конструирование. – М.: РУДН, 2007. – 259 с
70. Справочник по теории автоматического управления. / под ред. А.А. Красовского. – М.: Наука, 1987. – 712 с.
71. Афанасьев В.Н., Колмановский В.Б., Носов В.Р. Математическая теория конструирования систем управления. – М.: Высшая школа, 2003. – 573 с.
72. Boltyanskii V.G. Mathematical Methods of Optimal Control. – Holt, Rinehart and Winston, New York, 1971. – 272 p.

73. Егупов Н.Д., Пупков К.А. Методы классической и современной теории автоматического управления. Синтез регуляторов систем автоматического управления. В 5 т. – М.: МГТУ им. Баумана, 2004. – Т. 3. – 616 с.
74. Safonov M. G., Athans M. A multiloop generalization of the circle criterion for stability margin analysis // IEEE Trans. on Automatic Control. – 1981. – Vol. – 26. – no. 2. – pp. 415–422.
75. Boyd S., Ghaoui E., Feron E., Balakrishnan V. Linear matrix inequalities in systems and control theory. – Philadelphia: Society for Industrial and Applied Mathematics, 1994. – 193 p.
76. Tsypkin Ya.Z, Polyak B.T. Robust absolute stability of continuous systems // Int. J. Nonlin. Control. – 1993. – V. 3. – No. 3. – pp. 231-239.
77. Цыкунов А.М. Робастное управление нестационарными объектами //АиТ. – 1996. – №2. – С.117-123.
78. Furtat I.B. Robust control for a specific class of non-minimum phase dynamical networks. // J. Comput. Syst. Sci. Int. – 2014. – 53. – pp. 33–46.
79. Бесекерский В.А., Попов Е.П. Теория систем автоматического управления. – М.: Наука, 1975. – 767 с.
80. Ройтенберг Я.Н. Автоматическое управление. – 2-е изд., перераб. и доп. – Москва: Наука, 1978. – 551 с.
81. Румянцев В.В., Озиранер А.С. Устойчивость и стабилизация движения по отношению к части переменных. – М.: Наука, 1987. – 253 с.
82. Ким Д.П. Теория автоматического управления. Т.2. Многомерные, нелинейные, оптимальные и адаптивные системы. – М. : ФИЗМАТЛИТ, 2004. – 464 с
83. Коробов В.И. Управляемость, устойчивость некоторых нелинейных систем // Дифференц.уравнения. – 1973. – Т.9. – Вып. 4. – С. 614–619.
84. Колмогоров А.Н., Фомин С.В. Элементы теории функции и функционального анализа – 7-е изд. – М.: ФИЗМАТЛИТ, 2004. – 572 с.
85. Ljung L. System Identification: Theory for the User (second ed.). – Upper Saddle River, New Jersey: Prentice-Hall, 1999.

86. Dastangoo P., Ramirez-Serrano A. Non-linear Parameter Identification for Humanoid Robot Components // The 7th International Conference of Control, Dynamic Systems, and Robotics. – 2020. 10.11159/cdsr20.148.
87. Алексеев А.А., Кораблев Ю.А., Шестопалов М.Ю. Идентификация и диагностика систем: учеб. для студ. высш. учеб. заведений – М.: Издательский центр «Академия», 2009.
88. Cox P., Toth R. Linear parameter-varying subspace identification: A unified framework // Automatica. – 2021. – 123. – 109296.
89. Sjöberg J., Zhang Q., Ljung L., Benveniste A., Delyon B., Glorennec P., Hjalmarsson H., Juditsky A. Nonlinear black-box modeling in system identification: a unified overview // Automatica. – 1995. – 31(12). – pp. 1691–1724.
90. Nelles O. Classical Polynomial Approaches. // In: Nonlinear System Identification. – Springer, Berlin, Heidelberg, 2001.
91. Fakhrizadeh Esfahani A., Dreesen P., Tiels K., Noël J.-P., Schoukens J. Parameter reduction in nonlinear state-space identification of hysteresis. // Mechanical Systems and Signal Processing. – 2017. – 104. 10.1016/j.ymssp.2017.10.017.
92. Liu G. P. Nonlinear identification and control: a neural network approach. – Springer Science & Business Media, 2012.
93. Werbos P. J. (n.d.). Neural networks for control and system identification. // Proceedings of the 28th IEEE Conference on Decision and Control. – 1989. doi:10.1109/cdc.1989.70114
94. Fu Z. J. et al. Nonlinear systems identification and control via dynamic multitime scales neural networks // IEEE Transactions on neural networks and learning systems. – 2013. – V.24. – №.11. – pp. 1814-1823.
95. Шмалько Е.Ю., Румянцев Ю.А., Байназаров Р.Р., Ямшанов К.Л. Идентификация нейросетевой модели робота для решения задачи оптимального управления. // Информатика и автоматизация. – 2021. – 20(6). – С. 1254-1278.

96. Шмалько Е.Ю., Прокопьев И.В., Дивеев А.И. Поддержка средств автономной навигации мобильного робота с помощью внутренней модели на нейронной сети // International Journal of Open Information Technologies. – 2023. – Т. 11. – № 2. – С. 25-31.
97. Flores J.J., Graff M. System Identification Using Genetic Programming and Gene Expression Programming. // In: Yolum, p., Güngör, T., Gürgen, F., Özturan, C. (eds) Computer and Information Sciences – ISCIS 2005. Lecture Notes in Computer Science. – 2005. – Vol 3733. – Springer, Berlin, Heidelberg. – pp. 503-511.
98. Дивеев А.И., Софронова Е.А., Шмалько Е.Ю. Метод идентификационного синтеза управления и его применение к мобильному роботу // Информационные и математические технологии в науке и управлении. 2016. № 2. С. 53-61.
99. Sebag M. Genetic Programming Applied to Model Identification. // In: Györfi, L. (eds) Principles of Nonparametric Learning. International Centre for Mechanical Sciences. – 2002. – Vol 434. – Springer, Vienna. – pp. 271–335.
100. Walsh G., Tilbury D., Sastry S., Murray R., Laumond J.P. Stabilization of trajectories for systems with nonholonomic constraints. // IEEE Trans. Autom. Control – 1994. – 39. – pp. 216-222.
101. Samir A., Hammad A., Hafez A., Mansour H. Quadcopter Trajectory Tracking Control using State-Feedback Control with Integral Action. // Int. J. Comput. Appl. – 2017. – 168. – pp.1-7.
102. Allagui N.Y., Abid D.B., Derbel N. Autonomous navigation of mobile robot with combined fractional order PI and fuzzy logic controllers. // In Proceedings of the 2019 16th International Multi-Conference on Systems, Signals and Devices (SSD). – Istanbul, Turkey, 21-24 March 2019. – pp. 78-83.
103. Chen B., Cao Y., Feng Y. Research on Trajectory Tracking Control of Non-holonomic Wheeled Robot Using Backstepping Adaptive PI Controller. // In Proceedings of the 2022 7th Asia-Pacific Conference on Intelligent Robot Systems (ACIRS). – Tianjin, China, 1-3 July 2022. – pp. 7-12.

104. Karnani C., Raza S., Asif A., Ilyas M. Adaptive Control Algorithm for Trajectory Tracking of Underactuated Unmanned Surface Vehicle (UUSV). // Journal of Robotics. – 2023. – Volume 2023. – Article ID 4820479.

105. Nguyen A.T., Nguyen X.-M., Hong S.-K. Quadcopter Adaptive Trajectory Tracking Control: A New Approach via Backstepping Technique. // Appl. Sci. – 2019. – 3873. – pp. 1-17.

106. Павлов И.П. Лекции о работе больших полушарий головного мозга. – М.: Эксмо, 2016. – 480 с.

107. Diveev A, Shmalko E. Research of Trajectory Optimization Approaches in Synthesized Optimal Control // Symmetry. – 2021. – 13(2): 336.

108. Diveev A., Shmalko E. Adaptive Synthesized Control for Solving the Optimal Control Problem. // Mathematics. – 2023. – 11. – 4035.

109. Дивеев А.И. Расширенная задача оптимального управления и численный метод ее решения. // Мехатроника, автоматизация, управление. 2024 – 25(3). – С. 111-120.

110. Shmalko E., Diveev A. Extended Statement of the Optimal Control Problem and Machine Learning Approach to Its Solution. // Mathematical Problems in Engineering. – 2022. – Volume 2022. – Article ID 1932520.

111. Shmalko E., Diveev A. Additional Requirement in the Formulation of the Optimal Control Problem for Applied Technical Systems. // Engineering Proceedings. – 2023. – 33(1):7.

112. Ляпунов А.М. Общая задача об устойчивости движения – Москва; Ленинград: Гос. изд-во техн.-теорет. лит., 1950. – 472 с.

113. Зубов В.И. Устойчивость движения. Методы Ляпунова и их применение. Учеб. пособие для мех.-мат. спец. ун-тов. – 2-е изд., перераб. и доп. – М.: Высш. шк., 1984. – 232 с.

114. Демидович Б. П. Лекции по математической теории устойчивости. – Москва: Наука, 1967. – 472с.

115. Четаев Н. Г. Устойчивость движения: Учеб. руководство. – 4е изд., испр. – М.: Наука. Гл. ред. физ.-мат. лит., 1990 – 176 с.

116. Александров А.Ю., Платонов А.В. Метод сравнения и устойчивость движений нелинейных систем: монография. – СПб.: Изд-во С.-Петерб. ун-та, 2012. – 263 с.

117. Lakshmikantham V., Leela S., Martynyuk A.A. Stability analysis of nonlinear systems. – New York; Basel: Dekker, Cop., 1989. – 315 c.

118. Blondel V., Gevers M., Lindquist A. Survey on the state of the systems and control // European J. Contol. – 1995. – Vol. 1. – pp. 5–23.

119. Syrmos V. L., Abdallah C. T., Dorato P., Grigoriadis K. Static Output Feedback. A Survey // Automatica. – 1997. – Vol. 33. – No. 2. – pp. 125–137.

120. Поляк Б. Т., Щербаков П. С. Трудные задачи линейной теории управления. Некоторые подходы к решению // Автоматика и телемеханика. – 2005. – № 5. – С. 7–46.

121. Brockett R. A stabilization problem. // In: Blondel, V., Sontag, E.D., Vidyasagar, M., Willems, J.C. (eds) Open Problems in Mathematical Systems and Control Theory. Communications and Control Engineering. – Springer, London, 1999. – pp. 75–78.

122. Леонов Г. А. Проблема Брокетта в теории устойчивости линейных дифференциальных уравнений // Алгебра и анализ. – 2001. – Т. 13. – № 4. – С. 134–155.

123. Moreau L., Aeyels D. Periodic output feedback stabilization of single-input single-output continuous-time systems with odd relative degree // Systems and Control Letters. – 2004. – Vol. 51. – No. 5. – pp. 395–406.

124. Мирошник И.В., Никифоров В.О., Фрадков А.Л. Нелинейное и адаптивное управление сложными механическими системами. – СПб.: Наука, 2000. – 548 с.

125. Бобцов А.А., Лямин А.В., Сергеев К.А. Синтез закона адаптивного управления для стабилизации не точно заданных нестационарных объектов // Изв. ВУЗов. Приборостроение. – 2001. – №3. – С. 3-7.

126. Теория автоматического регулирования. Книга 1. Математическое описание, анализ устойчивости и качества систем автоматического

регулирования (под ред. В.В. Солодовникова). – М.: Машиностроение, 1967. – 770 с.

127. Филипс Ч., Харбор Р. Системы управления с обратной связью. – М.: ЛБЗ, 2001. – 616 с.

128. Методы классической и современной теории автоматического управления. Т.3. Методы современной теории автоматического управления / Под ред. Н.Д. Егупова. – М.: МВТУ, 2000. – 748 с.

129. Григорьев В.В., Бойков В.И., Парамонов А.В., Быстров С.В., Проектирование регуляторов систем управления – СПб: Университет ИТМО, 2021. – 94 с.

130. Simon J.D.; Mitter S.K. A theory of modal control. // Inf. Control. – 1968. – 13. – pp.316 –353.

131. Бобцов А.А., Пыркин А.А., Фуртат И.Б., Управление системами с запаздыванием. Учебное пособие. – СПб.: Университет ИТМО, 2014. – 120с.

132. Оптимальная стабилизация линейных систем: Учеб. пособие / Тамасян Г.Ш., Фоминых А. В. – СПб.: Изд-во ВВМ, 2022. – 66 с.

133. Егоров А.И. Уравнение Риккати. – М.: Физматлит, 2001. – 320 с.

134. Kokotovic, P.V. The joy of feedback: nonlinear and adaptive // IEEE Control Systems Magazine. – 1992. – 12 (3). – pp 7–17.

135. Hassan K. Khalil. Nonlinear Systems (3rd Edition). – New Jersey: Prentice Hall, 2002. – 750 p.

136. Huang H., Gao J. Backstepping and Novel Sliding Mode Trajectory Tracking Controller for Wheeled Mobile Robots. // Mathematics. – 2024. – 12. – 1458. https://doi.org/10.3390/math12101458

137. Колесников А.А., Колесников А.А., Кузьменко А.А. Методы АКАР и АКОР в задачах синтеза нелинейных систем управления. // Мехатроника, автоматизация, управление. – 2016. – 17(10) – pp. 657-669.

138. Колесников А.А. Синергетическая теория управления – Таганрог: ТРГУ, М.: Энергоатомиздат, 1994. – 344 с.

139. Кунцевич В. М., Лычак М. М. Синтез систем автоматического управления с помощью функций Ляпунова. – М.: Наука, 1977. – 400 с.

140. Li Y., Cai Y., Wang Y., Li W., Wang G. Simultaneous Tracking and Stabilization of Nonholonomic Wheeled Mobile Robots under Constrained Velocity and Torque. // Mathematics. – 2024. – 12. – 1985. https://doi.org/10.3390/math12131985

141. Diveev A., Shmalko E. Machine Learning Control by Symbolic Regression. – Springer, Cham, 2021. – 155 p.

142. Duriez T.; Brunton S.L.; Noack, B.R. Machine Learning Control - Taming Nonlinear Dynamics and Turbulence. – Springer: Berlin, Heidelberg, Germany, 2017. – 211p.

143. Cornejo Maceda, G.Y., Noack, B.R. Evolutionary Machine Learning in Control. // In: Banzhaf, W., Machado, P., Zhang, M. (eds) Handbook of Evolutionary Machine Learning. Genetic and Evolutionary Computation. – Springer, Singapore, 2024. – pp. 629–656.

144. Shmalko, E., Diveev, A. Control Synthesis as Machine Learning Control by Symbolic Regression Methods // Applied Sciences. – 2021. – 11(12): 5468.

145. Brunton S.L., Kutz J.N. Data-Driven Science and Engineering: machine learning, dynamical systems and control. – Cambridge University Press, 2019. – 492 c.

146. Michalewicz Z. Krawczyk J., Kazemi M., Janikow C.Z. Genetic algorithms and optimal control problems. // Proceedings of the IEEE Conference on Decision and Control. – 1991. – 3. – pp. 1664 – 1666.

147. Romera-Paredes B., Barekatain M., Novikov A. et al. Mathematical discoveries from program search with large language models // Nature. – 2023. https://doi.org/10.1038/s41586-023-06924-6

148. Izzo D., Sprague C.I., Tailor D.V. Machine Learning and Evolutionary Techniques in Interplanetary Trajectory Design. // In: Fasano, G., Pintér, J. (eds) Modeling and Optimization in Space Engineering. Springer Optimization and Its Applications, vol 144. – Springer, Cham, 2019. – pp. 191–210.

149. Sánchez-Sánchez C., Izzo D., Hennes D. Learning the optimal state-feedback using deep networks // 2016 IEEE Symposium Series on Computational Intelligence (SSCI). – Athens, Greece, 2016. – pp. 1-8.

150. Константинов С.В. Решение задачи синтеза системы управления на основе аппроксимации множества оптимальных траекторий методом сетевого оператора: дис. канд. техн. наук: 2.3.1. – ФИЦ ИУ РАН, Москва, 2022. – 180 с.

151. Fleming P.J., Purshouse R.C. Evolutionary algorithms in control systems engineering: A survey. // Control Eng. Pract. – 2002. – 10. – pp. 1223-1241.

152. Gage P., Braun R., Kroo I. Interplanetary trajectory optimization using a genetic algorithm. // Journal of the Astronautical Sciences. – 1995. – 43(1). – pp. 59–76.

153. Vasile M., Minisci E., Locatelli M. Analysis of some global optimization algorithms for space trajectory design. // Journal of Spacecraft and Rockets. – 2010. – 47(2): 334.

154. Di Lizia P., Radice G. Advanced global optimisation for mission analysis and design. Final Report Ariadna id 04/4101, 2014.

155. Izzo D., Becerra V.M., Myatt D.R., Nasuto S.J., Bishop J.M. Search space pruning and global optimisation of multiple gravity assist spacecraft trajectories. // Journal of Global Optimization. – 2007. – 38(2). – pp.283–296.

156. Liu X., Jiang D., Tao B., Jiang G., Sun Y., Kong J., Tong X., Zhao G., Chen B. Genetic Algorithm-Based Trajectory Optimization for Digital Twin Robots. // Front. Bioeng. Biotechnol. – 2022. – 9: 793782.

157. Qiao T., Sambo Y.A., Imran M.A., Ahmad W. Drone Trajectory Optimization using Genetic Algorithm with Prioritized Base Stations // 2020 IEEE 25th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD). – Pisa, Italy, 2020. – pp. 1-6.

158. Zanchettin A.M., Messeri C., Cristantielli D. et al. Trajectory optimisation in collaborative robotics based on simulations and genetic algorithms. //

International Journal of Intelligent Robotics and Applications. – 2022. – Volume 6. – pp. 707–723.

159. Liu C., Cao Gh., Qu Yy. et al. An improved PSO algorithm for time-optimal trajectory planning of Delta robot in intelligent packaging. // The International Journal of Advanced Manufacturing Technology. – 2020. – 107. – pp. 1091–1099.

160. Pham Van Bach N., Dam Hai Q., Bui Trung T. Optimization of trajectory tracking control of 3-DOF translational robot use PSO method based on inverse dynamics control for surgery application // Journal of Vibroengineering. – Aug. 2021. – Vol.23. – No. 7. – pp. 1591–1601.

161. Stanovov V., Akhmedova S., Semenkin E. Solving the Global Trajectory Optimization Problem with Archive-Based Differential Evolution. // 2018 International Conference on Information Technologies (InfoTech). – Varna, Bulgaria. – 2018. – pp. 1-3.

162. Giri R., Ghose D. Differential Evolution Based Ascent Phase Trajectory Optimization for a Hypersonic Vehicle. // In: Panigrahi, B.K., Das, S., Suganthan, P.N., Dash, S.S. (eds) Swarm, Evolutionary, and Memetic Computing. SEMCCO 2010. Lecture Notes in Computer Science, vol 6466. – Springer, Berlin, Heidelberg, 2010. – pp. 11–18.

163. Митрофанов С.А. Оптимизация траектории полета космического аппарата методом дифференциальной эволюции // Актуальные проблемы авиации и космонавтики. – 2018. – №14. – С.81-83.

164. Radice G., Olmo G. Ant colony algorithms for two-impulse interplanetary trajectory optimization. // Journal of Guidance Control and Dynamics. – 2006. – 29(6): 1440.

165. Brand M., Masuda M., Wehner N., Xiao-Hua Yu. Ant Colony Optimization algorithm for robot path planning // 2010 International Conference On Computer Design and Applications. – Qinhuangdao, China. – 2010. – pp. V3-436 – V3-440.

166. Dario Izzo, Luís F. Simões, Marcus Märtens, Guido C.H.E. de Croon, Aurelie Heritier, and Chit Hong Yam. Search for a grand tour of the Jupiter galilean moons. // In Proceedings of the 15th annual conference on Genetic and evolutionary computation (GECCO '13). Association for Computing Machinery, New York, USA. – 2013. – pp. 1301–1308.

167. Wolpert D.H., Macready W.G. No free lunch theorems for optimization. // IEEE Transactions on Evolutionary Computation. – April 1997 – vol. 1. – no. 1. – pp. 67-82.

168. Nayyar A., Le D.-N., Nguyen N.G. Advances in Swarm Intelligence for Optimizing Problems in Computer Science (1st ed.). – New York: Chapman and Hall/CRC. – 2018. – 314 p.

169. Гладков Л.А., Курейчик В.В., Курейчик В.М. Генетические алгоритмы / Под ред. В.М. Курейчика. – 2-е изд. – М.: ФИЗМАТЛИТ, 2006. – 320 с.

170. Карпенко А. П., Селиверстов Е.Ю. Обзор методов роя частиц для задачи глобальной оптимизации (particle swarm optimization) // Наука и образование. МГТУ им. Н.Э. Баумана: электрон. журн. – 2009. – № 3. – С. 2-26.

171. Kennedy J., Eberhart R. Particle Swarm Optimization. // Proceedings of IEEE International Conference on Neural Networks. – 1995. – Vol. IV. – pp. 1942–1948.

172. Zhang Y. A Comprehensive Survey on Particle Swarm Optimization Algorithm and Its Applications. // Mathematical Problems in Engineering. – 2015. – 931256.

173. Skanderova L. Self-organizing migrating algorithm: review, improvements and comparison // Artificial Intelligence Review. – 2023. – Vol. 56. – pp. 101–172.

174. Diveev A., Sofronova E., Shmalko E. Modified SOMA for Optimal Control Problem. // 2019 IEEE Congress on Evolutionary Computation (CEC). – Wellington, New Zealand. – 2019. – pp. 2894-2899.

175. Faris H., Aljarah I., Al-Betar M.A. et al. Grey wolf optimizer: a review of recent variants and applications // Neural Comput. and Applic. – 2018. – Vol.30. – pp. 413–435.

176. Diveev A., Shmalko E. Machine Learning Feedback Control Approach Based on Symbolic Regression for Robotic Systems // Mathematics. – 2022. – 10(21), 4100.

177. Deisenroth M.P., Faisal A.A., Ong C.S. Mathematics for Machine Learning. – Cambridge University Press: Cambridge, UK, 2020. – 398 p.

178. Burkov A. The Hundred-Page Machine Learning Book. – Andriy Burkov, Quebec City, QC, Canada, 2019. – 160p.

179. Géron A. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. – O'Reilly Media, Inc.: Sebastopol, CA, USA, 2019. – 856p.

180. Brunton S.L., Proctor J.L., Kutz J.N. Discovering governing equations from data: Sparse identification of nonlinear dynamical systems. // Proc. Natl. Acad. Sci., USA. – 2015. – 113 (15). – pp. 3932–3937.

181. Savchenko A.V. Probabilistic Neural Network With Complex Exponential Activation Functions in Image Recognition // IEEE Transactions on Neural Networks and Learning Systems. – 2020. – vol. 31. – no. 2. – pp. 651-660.

182. Zhang W., Wang J., Lan F. Dynamic hand gesture recognition based on short-term sampling neural networks // IEEE/CAA Journal of Automatica Sinica. – 2021. – vol. 8. – no. 1. – pp. 110-120.

183. Жилов Р.А. Постройка ПИД-регулятора с использованием нейронных сетей // Известия Кабардино-Балкарского научного центра РАН. – 2022. – № 5 (109). – С. 38–47.
DOI: 10.35330/1991- 6639-2022-5-109-38-47

184. Shmalko E., Diveev A. Control Synthesis as Machine Learning Control by Symbolic Regression Methods // Applied Sciences. – 2021. – 11. – 5468.

185. Alibekov E., Kubalık J., Babuska R. Symbolic Method for Deriving Policy in Reinforcement Learning // Proceedings 2016 IEEE 55th Conference on Decision and Control (CDC). – December 12-14, 2016, Las Vegas, USA. – pp.2789-2795.

186. Колмогоров А.Н. О представлении непрерывных функций нескольких переменных суперпозициями непрерывных функций меньшего числа переменных // Докл. АН СССР – 1956. – Т. 108 – № 2. – С.179–182.

187. Kolmogorov A.N. On the Representation of Continuous Functions of Many Variables by Superposition of Continuous Functions of One Variable and Addition // American Math. Soc. Transl. – 1963. – 28. – pp. 55-63.

188. Арнольд В.И. О функциях трех переменных // Докл. АН СССР – 1957. – 114:4. – С. 679–681

189. Holland J. Adaptation in Natural and Artificial Systems. – Cambridge, MA: MIT Press, 1992. – 232 p.

190. Mitchell M. An Introduction to Genetic Algorithms. – Cambridge, MA: MIT Press, 1996. – 221 p.

191. Goldberg D. Genetic Algorithms in Search, Optimization and Machine Learning. – Reading, MA: Addison-Wesley Professional, 1989. – 412 p.

192. Vose M. The Simple Genetic Algorithm: Foundations and Theory. – Cambridge, MA: MIT Press, 1999. – 251 p.

193. Kumar M., Husian M., Upreti N., Gupta D. Genetic Algorithm: review and application // Journal of Information and Knowledge Management. – 2010. – 2. – pp. 451-454.

194. Diveev A.I. Small Variations of Basic Solution Method for Non-numerical Optimization // IFAC-PapersOnLine. – 2015. – vol. 48(25). – pp.028–033.

195. Дивеев А.И. Метод сетевого оператора. – М.: Изд-во ВЦ РАН, 2010. – 178 с.

196. Koza J.R., Keane M.A., Yu J., Bennett F.H., MydlowecW., Stiffelman O. Automatic synthesis of both the topology and parameters for a robust controller for a non-minimal phaseplant and a three-lag plant by means of genetic programming // Proceedings of IEEE Conference on Decision and Control. – 1999. – pp. 5292–5300.

197. Koza J. R., Bennett I. F. H., Andre D., Keane M. A., Dunlap F. Automated synthesis of analog electrical circuits by means of genetic programming // IEEE Transactions on Evolutionary Computation. – 1997. – 1(2). – pp. 109–128.

198. Derner E., Kubalík J., Ancona N., Babuška R. Symbolic Regression for Constructing Analytic Models in Reinforcement Learning // Applied Soft Computing. – September 2020. – Vol. 94. – pp. 1 –12.

199. Brunton S. L., Noack B. R.Closed-loop turbulence control: Progress and challenges // Applied Mechanics Reviews. – 2015. – vol. 67. – no. 5. – 050801.

200. Cornejo Maceda G. Y., Li Y., Lusseyran F., Morzynski M., Noack B. R. Stabilization of the fluidic pinball with gradient-enriched machine learning control // J. Fluid Mech. – 2021. – vol. 917. – A42.

201. Дивеев А.И., Шмалько Е.Ю. Многокритериальный структурно-параметрический синтез системы управления спуском космического аппарата на основе метода сетевого оператора // Вестник Российского университета дружбы народов. Серия инженерные исследования. – 2008. – №4. – С. 86–93.

202. Дивеев А.И., Шмалько Е.Ю. Метод автоматического подбора формул для синтеза системы управления спуском космического аппарата // Труды института Системного анализа РАН. Динамика неоднородных систем – М.: ИСА РАН, КомКнига, 2008. – Т 32(1). – С. 7–15.

203. Программный пакет расчета значений управляющих сигналов по матрице сетевого оператора [Электронный ресурс]: https://github.com/urock/network_operator

204. Дивеев А.И., Шмалько Е.Ю. Синтез системы автоматического управления мобильным роботом методом сетевого оператора и

алгоритмом интеллектуальной эволюции // Нелинейный мир. ЗАО «Издательство «Радиотехника». – 2014. – №7, т.12. – С.42-48.

205. Diveev A.I., Shmalko E.Yu. Self-adjusting Control for Multi Robot Team by the Network Operator Method // 2015 European Control Conference (ECC) – July 15-17, 2015. – Linz, Austria. – pp. 709-714.

206. Diveev A. I., Shmalko E. Yu. Automatic Synthesis of Control for Multi-Agent Systems with Dynamic Constraints // IFAC-PapersOnLine. – 2015. – Volume 48. – Issue 11. – pp. 384 – 389.

207. Koza J.R. Genetic Programming: On the Programming of Computers by Means of Natural Selection. – MIT Press, Cambridge, Massachusetts, London, MA, 1992. – 819 p.

208. Diveev A.I., Ibadulla S.I., Konyrbaev N.B., Shmalko E.Yu. Variational Genetic Programming for Optimal Control System Synthesis of Mobile Robots // IFAC-PapersOnLine. – 2015. – Volume 48. – Issue 19. – pp. 1-286.

209. Дивеев А.И., Шмалько Е.Ю. Программный комплекс для решения задачи синтеза управления методом вариационного генетического программирования: Свидетельство о государственной регистрации программы для ЭВМ №2015617366. От 08 июля 2015 г.

210. Zelinka I. Analytic programming by Means of SOMA Algorithm. // In Proceedings of 8th International Conference on Soft Computing Mendel 02. – Brno, Czech Republic, 2002. – pp. 93–101.

211. Zelinka I., Oplatkova Z., Nolle L. Analytic Programming – Symbolic Regression by Means of Arbitrary Evolutionary Algorithms // Transactions of The Society for Modeling and Simulation International. – 2005. – 6(9). – pp.1473-8031

212. Дивеев А.И., Шмалько Е.Ю. Метод вариационного аналитического программирование для синтеза системы управления: Свидетельство о государственной регистрации программы для ЭВМ №2014615476. От 27 мая 2014 г.

213. Diveev A.I., Ibadulla S.I., Konyrbaev N.B., Shmalko E.Yu. Variational Analytic Programming for Synthesis of Optimal Control for Flying Robot // IFAC-PapersOnLine. – 2015. – Volume 48. – Issue 19. – pp. 1-286.

214. Дивеев А.И., Шмалько Е.Ю. Метод бинарного генетического программирования для автоматизации поиска решения задачи синтеза управления // Вопросы теории безопасности и устойчивости систем. – 2017. – № 19. – С. 23-39.

215. Diveev A., Shmalko E. Complete binary variational analytic programming for synthesis of control at dynamic constraints // ITM Web of Conferences. – 2017. – Т. 10. – 02004.

216. Дивеев А.И., Шмалько Е.Ю. Программный комплекс для решения задачи стабилизации роботов методом бинарного вариационного генетического программирования: Свидетельство о государственной регистрации программы для ЭВМ №2017662486 от 09 ноября 2017 г.

217. Miller J., Thomson P. Cartesian Genetic Programming. // Proc. EuroGP'2000R 3rd European Conf. Genetic Programming; R. Poli, W. Banzhaf, W.B. Langdon, J.F. Miller, P. Nordin, and Fogarty, T.C. Eds., Edinburgh, Scotland. – Berlin: Springer-Verlag, 2000. – vol. 1802.– pp. 121-132.

218. Miller J.F. Cartesian Genetic Programming. Natural Computing Series, Springer, Berlin, Heidelberg, 2011. – 346 p.

219. Дивеев А.И., Шмалько Е.Ю. Программа для синтеза системы стабилизации на основе Вариационного Декартового Генетического Программирования: Свидетельство о государственной регистрации программы для ЭВМ №2021611899 от 08 февраля 2021 г.

220. Diveev A.I., Shmalko E.Yu. Machine-Made Synthesis of Stabilization System by Modified Cartesian Genetic Programming, // IEEE Transactions on Cybernetics. – July 2022. – vol. 52. – no. 7. – pp. 6627-6637.

221. Diveev A.I., Shmalko E.Yu. Optimal Motion Control for Multi-Robot System by Multilayer Network Operator // Proceedings of the 11th IEEE

Conference on Industrial Electronics and Applications (ICIEA 2016), 5 - 7 June 2016, Hefei, China. – pp. 2168-2173.

222. Diveev A.I., Shmalko E.Yu. Optimal Control Synthesis for Group of Robots by Multilayer Network Operator // Proceedings of the International conference on Control, Decision and Information technologies 2016, CoDIT-2016. – Saint Julian's, Malta, 2016. – pp. 077-082.

223. Дивеев А.И., Шмалько Е.Ю. Синтез управления для автономной группы роботов с фазовыми ограничениями методом многослойного сетевого оператора с расстановкой приоритетов // Вестник Российского университета дружбы народов. Серия: Инженерные исследования. – 2017. – Т. 18. – № 1. – С. 115-124.

224. Дивеев А.И., Шмалько Е.Ю. Метод многослойного сетевого оператора в задаче синтеза системы управления группой роботов // Труды Второй молодежной научной конференции «Задачи современной информатики» – М.: ФИЦ ИУ РАН, 2015. – С. 241-247.

225. Дивеев А.И., Шмалько Е.Ю. Программа автоматического синтеза генератора оптимальных траекторий для группы роботов методом многослойного сетевого оператора: Свидетельство о государственной регистрации программы для ЭВМ №2017662485 от 09 ноября 2017 г

226. Šuster P., Jadlovská A. Tracking Trajectory of the Mobile Robot Khepera II Using Approaches of Artificial Intelligence // Acta Electrotechnica et Informatica. – 2011. – Vol. 11. – No. 1. – pp. 38–43.

227. Лавренов Р.О., Магид Е.А., Мацуно Ф., Свинин М.М., Сутакорн Д. Разработка и имплементация сплайн-алгоритма планирования пути в среде ROS/Gazebo. // Труды СПИИРАН. – 2019. – 18(1). – С. 57-84.

228. Zhang B., Liu P. Control and benchmarking of a 7-DOF robotic arm using Gazebo and ROS // Peer J Computer Science. – 2021. – 7. – e383.

229. Описание симуляционного робота Rosbot [Электронный ресурс]: https://github.com/husarion/rosbot_description

230. Дивеев А.И., Шмалько Е.Ю. Метод синтезированного оптимального управления для группы роботов // Надежность и качество сложных систем. – 2018. – № 4 (24). – С. 40–47.

231. Diveev, A., Shmalko, E., Serebrenny, V., Zentay, P. Fundamentals of synthesized optimal control // Mathematics. – 2021. – 9(1). – pp. 1–18.

232. Shmalko E., Diveev A. Synthesized Optimal Control for Mecanum-wheeled Robot // Proceedings 8th International Conference on Control, Decision and Information Technologies, CoDIT 2022, Istanbul, Turkey. – 2022. – pp. 599-604.

233. Шмалько Е.Ю. Синтезированное оптимальное управление для мобильного робота с mecanum-колесами // Вопросы теории безопасности и устойчивости систем. – 2022. – № 24. – С. 83-97.

234. Шмалько Е.Ю. Система управления квадрокоптером на основе принципа синтезированного оптимального управления // Труды международного симпозиума «Надежность и качество». – 2023. – Т. 1. – С. 81-84.

235. Гурьянов А.Е. Моделирование управления квадрокоптером // Инженерный вестник. МГТУ им. Н.Э. Баумана. – 2014. – №08. – С. 522-534.

236. Kingma D., Ba J. Adam: A Method for Stochastic Optimization // International Conference on Learning Representations. – 2014. – abs/1412.6980.

Публикации автора по теме диссертации

Монография

1. *Diveev A., Shmalko E.* Machine Learning Control by Symbolic Regression. Springer, Cham, 2021. – 155 p. **(Scopus)**

В изданиях из списка ВАК РФ

2. *Шмалько Е.Ю.* Машинно синтезированное управление нелинейным динамическим объектом на основе оптимального расположения точек равновесия // Информатика и автоматизация (Труды СПИИРАН). – 2023. – Т. 22. – № 1. – С. 87-109. **(К-1, RSCI)**
3. *Шмалько Е.Ю.* Машинное обучение систем управления с обратной связью на базе принципа синтезированного оптимального управления // Надежность и качество сложных систем. – 2023. – № 3 (43). – С. 38-49. **(К-2)**
4. *Дивеев А.И., Шмалько Е.Ю, Хуссейн О.* Синтезированное оптимальное управление групповым взаимодействием квадрокоптеров на основе многоточечной стабилизации // Вестник МГТУ им. Н.Э. Баумана, Серия Приборостроение. – 2020. – № 4. – Т.133. – С. 114-133. **(К-2, RSCI)**
5. *Дивеев А.И., Шмалько Е.Ю., Хуссейн О.* Управление квадрокоптером методом сетевого оператора на основе многоточечной стабилизации // Мехатроника, автоматизация, управление. – 2020. – № 7. – Т.21. – С. 428-438. **(К-1, RSCI)**
6. *Дивеев А.И., Шмалько Е.Ю.* Исследование синтезированного оптимального управления группой роботов при наличии неопределенностей // Надежность и качество сложных систем. – 2020. – № 2. – Т.30. – С. 10-18. **(К-2)**
7. *Дивеев А.И., Шмалько Е.Ю.* К практической реализации решения задачи оптимального управления // Надежность и качество сложных систем. – 2020. – № 2. – Т.30. – С. 37-45. **(К-2)**

8. *Дивеев А.И., Шмалько Е.Ю.* Метод синтезированного оптимального управления для группы роботов // Надежность и качество сложных систем. – 2018. – № 4 (24). – С.40-47. **(К-2)**
9. *Дивеев А.И., Шмалько Е.Ю.* Численные методы синтеза синергетического управления групповым взаимодействием роботов // «Известия ЮФУ. Технические науки». – 2017. – № 9. – С.6-21. **(К-2)**
10. *Дивеев А.И., Шмалько Е.Ю.* Эволюционные методы вычислений для синтеза управления группой роботов и поиска оптимальных траекторий их движения // Cloud of Science. – 2017. – Т. 4. – № 3. – С. 395-414. **(К-3)**
11. *А.И. Дивеев, Е.Ю. Шмалько* Численный синтез системы управления группой роботов методом символьной регрессии // Известия ЮФУ. Технические науки. – 2015 – № 10 (171). – С.29-45. **(К-2)**
12. *Дивеев А.И., Шмалько Е.Ю.* Двухэтапный синтез системы управления методом сетевого оператора // Вестник Российского университета дружбы народов, серия Инженерные исследования. – 2015. – №1. – С.91-100. **(К-3)**
13. *Дивеев А.И., Шмалько Е.Ю.* Синтез системы автоматического управления мобильным роботом методом сетевого оператора и алгоритмом интеллектуальной эволюции // Нелинейный мир, ЗАО «Издательство «Радиотехника» – 2014. – №7. – т.12. – С.42-48. **(К-2)**

В изданиях, входящих в базы цитирования Scopus и Web of Science

14. *Shmalko E.* Computational Approach to Optimal Control in Applied Robotics // In: Ronzhin, A., Pshikhopov, V. (eds) Frontiers in Robotics and Electromechanics. Smart Innovation, Systems and Technologies. Springer, Singapore – 2023. – vol. 329. – pp. 387-401. **(Scopus)**
15. *Shmalko E., Diveev A.* Machine Learning Control Synthesis by Symbolic Regression for Avoidance of Arbitrary Positioned Obstacles // 2023 9th International Conference on Control, Decision and Information Technologies (CoDIT), Rome, Italy – 2023. – pp. 668-673. **(Scopus)**

16. *Diveev, A., Shmalko, E.* Adaptive Synthesized Control for Solving the Optimal Control Problem // Mathematics. – 2023. – 11, 4035. **(Scopus, WoS, Q1)**
17. *Shmalko E., Diveev A.* Additional Requirement in the Formulation of the Optimal Control Problem for Applied Technical Systems. // Engineering Proceedings. – 2023. – 33(1):7. **(Scopus)**
18. *Diveev A. I., Shmalko E. Y.* Machine-Made Synthesis of Stabilization System by Modified Cartesian Genetic Programming. // IEEE Transactions on Cybernetics. – July 2022. – vol. 52. – no. 7. – pp. 6627-6637. **(Scopus, WoS, Q1)**
19. *Shmalko E., Diveev A.* Extended Statement of the Optimal Control Problem and Machine Learning Approach to Its Solution // Mathematical Problems in Engineering. – 2022. – vol. 2022. – Article ID 1932520. **(Scopus, WoS, Q2)**
20. *Diveev A., Shmalko E.* Machine Learning Feedback Control Approach Based on Symbolic Regression for Robotic Systems // Mathematics. – 2022. – 10(21), 4100. **(Scopus, WoS, Q1)**
21. *Diveev A. Shmalko E.* Stability of the Optimal Control Problem Solution // Proceedings 8th International Conference on Control, Decision and Information Technologies, CoDIT 2022, Istanbul, Turkey. – 2022. – pp. 33-38. **(Scopus, WoS)**
22. *Shmalko E., Diveev A.* Synthesized Optimal Control for Mecanum-wheeled Robot // Proceedings 8th International Conference on Control, Decision and Information Technologies, CoDIT 2022, Istanbul, Turkey. – 2022. – pp. 599-604. **(Scopus, WoS)**
23. *Shmalko E.* Feasibility of Synthesized Optimal Control Approach on Model of Robotic System with Uncertainties. // In: Ronzhin A., Shishlakov V. (eds) Electromechanics and Robotics. Smart Innovation, Systems and Technologies. – Springer, Singapore. – 2022. – vol 232. – pp.131-143. **(Scopus)**
24. *Shmalko E., Diveev A.* Control Synthesis as Machine Learning Control by Symbolic Regression Methods // Applied Sciences. – 2021. – 11(12): 5468. **(Scopus, WoS, Q2)**

25. *Diveev A, Shmalko E.* Research of Trajectory Optimization Approaches in Synthesized Optimal Control // Symmetry. – 2021. – 13(2): 336. **(Scopus, WoS, Q2)**

26. *Diveev A., Shmalko E.* Synthesized optimal control based on machine learning // Journal of Physics: Conference Series. – 2021. – 1727(1). – 012006. **(Scopus, Q4)**

27. *Diveev A., Shmalko E.* Comparative study of numerical solutions for the optimal control problem in the presence of uncertainties // Procedia Computer Science. – 2021. – 186. – pp. 279-286. **(Scopus, WoS)**

28. *Diveev A., Shmalko E., Serebrenny V., Zentay P.* Fundamentals of synthesized optimal control // Mathematics. – 2021. – 9(1). – pp. 1–18. **(Scopus, WoS, Q1)**

29. *Diveev A., Shmalko E.* Multi-point Stabilization Approach to the Optimal Control Problem with Uncertainties // Advances in Optimization and Applications. 11th International Conference, OPTIMA 2020 Moscow, Russia. – Springer Nature Switzerland AG, CCIS. – 2020. – 1340. – pp. 129-142. **(Scopus)**

30. *Diveev A., Shmalko E.* Optimal Control Design for a Group of Mobile Robots with Uncertainties // Proceedings of 2020 15th IEEE Conference on Industrial Electronics and Applications (ICIEA). – Kristiansand, Norway. – 2020. – pp.308-313. **(Scopus, WoS)**

31. *Diveev A., Hussein O., Shmalko E., Sofronova E.* Synthesis of Control System for Quad-Rotor Helicopter by the Network Operator Method // Intelligent Systems and Applications Proceedings of the 2020 Intelligent Systems Conference (IntelliSys), Volume 1, Springer Nature Switzerland AG 2021. Advances in Intelligent Systems and Computing. – 2021. – 1250 AISC. – pp.246-263. **(Scopus)**

32. *Diveev A., Shmalko E.* Optimal Feedback Control through Numerical Synthesis of Stabilization System // Proc. 7th International Conference on Control, Decision and Information Technologies (CoDIT'20). Prague, Czech Republic, June 29 - July 2, 2020. – pp. 112-117. **(Scopus, WoS)**

33. *Diveev, A., Shmalko, E.* Comparison of Direct and Indirect Approaches for Numerical Solution of the Optimal Control Problem by Evolutionary Methods // In:

Optimization and Applications. OPTIMA 2019. Communications in Computer and Information Science. – Springer, Cham. – 2020. – vol. 1145. **(Scopus, WoS, Q4)**

34. *Diveev, A.I., Sofronova, E.A., Shmalko, E.Y.* Modified SOMA for Optimal Control Problem. 2019 IEEE Congress on Evolutionary Computation (CEC). – 2019. – pp. 2894–2899. **(Scopus, WoS)**
35. *Diveev A., Sofronova E., Shmalko E.* A solution of synthesized optimal control problem for interaction of robots by evolutionary computations // Proceedings of the 14th IEEE Conference on Industrial Electronics and Applications, ICIEA. – 2019. – 14. – pp. 756–761. **(Scopus, WoS)**
36. *Diveev A.I., Shmalko E.Y., Sofronova E.A.* Theoretical fundamentals for unimodality estimation of an objective functional in the optimal control problem // 6th International Conference on Control, Decision and Information Technologies, CoDIT 2019. – 6. – pp. 767–772. **(Scopus, WoS)**
37. *Diveev A., Shmalko E.* Hybrid evolutionary algorithm for synthesized optimal control problem for group of interacting robots // 6th International Conference on Control, Decision and Information Technologies, CoDIT 2019. – 6. – pp.876–881. **(Scopus, WoS)**
38. *Diveev A.I., Shmalko E.Yu.* Evolutionary computations for synthesis of control system of group of robots and the optimum choice of trajectories for their movement // CEUR Workshop Proceedings of the VIII International Conference on Optimization and Applications (OPTIMA-2017), Petrovac, Montenegro, October 2-7, 2017. – pp. 158–165. **(Scopus)**
39. *Diveev A., Shmalko E.* Complete binary variational analytic programming for synthesis of control at dynamic constraints // ITM Web of Conferences. –2017. – T. 10. – 02004. **(Scopus, WoS)**
40. *Diveev A., Shmalko E.* Automatic approach to stabilization and control for multi robot teams by multilayer network operator // ITM Web of Conferences. – 2016. – Vol. 6. – 02004. **(Scopus, WoS)**
41. *Diveev A.I., Shmalko E.Yu.* Optimal Motion Control for Multi-Robot System by Multilayer Network Operator // Proceedings of the 11th IEEE Conference on

Industrial Electronics and Applications (ICIEA 2016), 5 - 7 June 2016, Hefei, China. – pp. 2168–2173. **(Scopus, WoS)**

42. *Diveev A. I., Shmalko E.Yu.* Optimal Control Synthesis for Group of Robots by Multilayer Network Operator // Proceedings of the International conference on Control, Decision and Information technologies 2016, CoDIT-2016, Malta, 6-8 April 2016. – pp.077–082. **(Scopus, WoS)**
43. *Diveev A.I., Ibadulla S.I., Konyrbaev N.B., Shmalko E.Yu.* Variational Analytic Programming for Synthesis of Optimal Control for Flying Robot // IFAC-PapersOnLine. – 2015. – Volume 48. – Issue 19. – pp. 75-80. **(Scopus, WoS, Q3)**
44. *A.I. Diveev, S.I. Ibadulla, N.B. Konyrbaev, E.Yu. Shmalko* Variational Genetic Programming for Optimal Control System Synthesis of Mobile Robots // IFAC-PapersOnLine. – 2015. – Volume 48. – Issue 19. – pp. 106-111. **(Scopus, WoS, Q3)**
45. *A.I. Diveev, E.Yu. Shmalko* Self-adjusting Control for Multi Robot Team by the Network Operator Method // 2015 European Control Conference (ECC) –July 15-17, 2015. Linz, Austria. – pp. 709-714. **(Scopus, WoS)**
46. *A.I. Diveev, E.Yu. Shmalko* Automatic Synthesis of Control for Multi-Agent Systems with Dynamic Constraints // IFAC-PapersOnLine. – 2015. – Volume 48. – Issue 11. – pp. 384-389. **(Scopus, WoS)**
47. *Diveev A., Khamadiyarov D., Shmalko E., Sofronova E.* Intellectual Evolution Method for Synthesis of Mobile Robot Control System // 2013 IEEE Congress on Evolutionary Computation, June 20-23, Cancun, Mexico. – pp. 24-31. **(Scopus, WoS)**

В трудах конференций и изданиях, входящих в РИНЦ

48. *Шмалько Е.Ю.* Проблема реализуемости решения задачи оптимального управления и принцип синтезированного управления к ее преодолению // В сборнике: XVI Всероссийская мультиконференция по проблемам управления (МКПУ-2023). Волгоград, 2023. – С. 100-101.

49. *Шмалько Е.Ю.* Учет неопределенности начального состояния при расчетах синтезированного оптимального управления в робототехнических системах // Прикладные проблемы системной безопасности: материалы Всероссийской конференции с межд. участием. Елец, 2023. – С.114-117.
50. *Шмалько Е.Ю., Ямшанов К.Л.* Реализация системы пространственной стабилизации как ROS-модуля для малогабаритного мобильного робота // Труды международного симпозиума «Надежность и качество». – 2023. – Т. 1. – С. 153-156.
51. *Шмалько Е.Ю.* Система управления квадрокоптером на основе принципа синтезированного оптимального управления // Труды международного симпозиума «Надежность и качество». – 2023. – Т. 1. – С. 81-84.
52. *Шмалько Е.Ю.* Синтезированное оптимальное управление для мобильного робота с mecanum-колесами // Вопросы теории безопасности и устойчивости систем. – 2022. – № 24. – С. 83-97.
53. *Шмалько Е.Ю., Серебренный В.В.* Применение методов машинного обучения для расчета синтезированного оптимального управления мобильным роботом // Труды 33-й международной научно-технической конференции "Экстремальная робототехника", Санкт-Петербург, 2022. – С.340-347.
54. *Румянцев Ю.А., Шмалько Е.Ю., Ямшанов К.Л.* Синтез контроллера обратной связи методом сетевого оператора для мобильного робота rosbot в иммитационной среде gazebo // Вопросы теории безопасности и устойчивости систем. – 2022. – № 24. – С. 98-109.
55. *Дивеев А.И., Шмалько Е.Ю.* Синтезированное управление для меканум робота // Труды международного симпозиума «Надежность и качество». – 2022. – т.1. – С. 27-29.
56. *Дивеев А.И., Шмалько Е.Ю.* К задаче машинного обучения управлению и методы ее решения // В сборнике: XIV Всероссийская мультиконференция по проблемам управления МКПУ-2021. Ростов-на-Дону, 2021. – С. 14-16.

57. *Шмалько Е.Ю.* Исследование вопроса оптимального расположения точек равновесия при синтезированном оптимальном управлении // Вопросы безопасности и устойчивости систем. – 2021. – № 23. – С. 29-39.

58. *Дивеев А.И., Шмалько Е.Ю.* Машинное обучение на основе символьной регрессии // В сборнике: Фундаментально-прикладные проблемы безопасности, живучести, надёжности, устойчивости и эффективности систем. материалы IV международной научно-практической конференции. Елец, 2020. – С. 191-195.

59. *Дивеев А.И., Шмалько Е.Ю.* Решение задачи оптимального управления для модели с возмущениями методом синтезированного оптимального управления // Вопросы теории безопасности и устойчивости систем. – 2020. – № 22. – С. 99-108.

60. *Дивеев А.И., Софронова Е.А., Шмалько Е.Ю.* Метод синтезированного оптимального управления групповым взаимодействием роботов. // Проектирование будущего. Проблемы цифровой реальности. – 2020. – 1(3). – С. 166-175.

61. *Дивеев А.И., Шмалько Е.Ю.* Синтезированное оптимальное управление для группы роботов // Материалы XII мультиконференции по проблемам управления (МКПУ-2019), 2019. – С. 138-140.

62. *Дивеев А.И., Шмалько Е.Ю.* Численное решение задачи оптимального управления группой роботов через синтез системы стабилизации // Фундаментально-прикладные проблемы безопасности, живучести, надежности, устойчивости и эффективности систем. Материалы III международной научно-практической конференции, 2019. – С. 259-264.

63. *Дивеев А.И., Шмалько Е.Ю.* Решение задачи группового управления с фазовыми ограничениями методом синтезированного оптимального управления // Вопросы теории безопасности и устойчивости систем. – 2019. – № 21. – С. 85-96.

64. *Дивеев А.И., Шмалько Е.Ю.* Классические методы символьной регрессии для поиска структур математических выражений (обзор) // Вопросы теории

безопасности и устойчивости систем. Изд. ФИЦ "Информатика и управление" РАН Москва. – 2018. – № 20 (20). – С. 100-132.

65. *Дивеев А.И., Шмалько Е.Ю.* Современные методы символьной регрессии и их модификации (обзор) // Вопросы теории безопасности и устойчивости систем. Изд. ФИЦ "Информатика и управление" РАН Москва. – 2018. – № 20 (20). – С. 133-158.

66. *Дивеев А.И., Шмалько Е.Ю.* Метод бинарного генетического программирования для автоматизации поиска решения задачи синтеза управления // Вопросы теории безопасности и устойчивости систем. – 2017. – № 19. – С. 23-39.

67. *Дивеев А.И., Шмалько Е.Ю.* Двухэтапный синтез систем управления групповым взаимодействием роботов методом символьной регрессии // Труды 10-й Всероссийской мультиконференции по проблемам управления (МКПУ-2017), 11 – 16 сентября 2017 г., с. Дивноморское, Геленджик, Россия. – С.276-278.

68. *Шмалько Е.Ю.* Синтез управления группой роботов численными методами с использованием MPI кластеров // Фундаментальные проблемы системной безопасности: материалы школы-семинара молодых ученых, посвященной 60-летию запуска первого в мире искусственного спутника Земли. – 2017. – С. 46-49.

69. *Дивеев А.И., Шмалько Е.Ю.* Синтез управления для автономной группы роботов с фазовыми ограничениями методом многослойного сетевого оператора с расстановкой приоритетов // Вестник Российского университета дружбы народов. Серия: Инженерные исследования. – 2017. – Т. 18. – № 1. – С. 115-124.

70. *Дивеев А.И., Шмалько Е.Ю., Рындин Д.А.* Решение задачи оптимального управления группой роботов эволюционными алгоритмами // Информационные и математические технологии в науке и управлении. – 2017. – № 3. – С. 109-121.

71. *Дивеев А.И., Софронова Е.А., Шмалько Е.Ю.* Эволюционные численные методы решения задачи синтеза системы управления группой роботов // Информационные и математические технологии в науке и управлении. – Иркутск: ИСЭМ СО РАН, 2016. – №3. – С. 11- 24.

72. *Дивеев А.И., Шмалько Е.Ю.* Синтез генератора оптимальных траекторий движения группы мобильных роботов методом многослойного сетевого оператора // Вопросы теории безопасности и устойчивости систем. – 2016. – № 18. – С. 32-41.

73. *Шмалько Е.Ю.* Методы эволюционных вычислений для решения сложных числовых и нечисловых задач // Фундаментальные проблемы системной безопасности: материалы III школы-семинара молодых ученых: в 2 частях. 2016. – С. 74-79.

74. *Дивеев А.И., Шмалько Е.Ю.* Повышение надежности систем управления группой объектов за счет автоматизации процесса их синтеза // Труды межд. симпозиума Надежность и качество. – 2016. – № 1. – С. 160-163.

75. *Дивеев А.И., Шмалько Е.Ю.* Методы генетического программирования для решения задачи синтеза оптимального управления // Вопросы теории безопасности и устойчивости систем. – 2015. – № 17. – С. 38-63.

76. *Дивеев А.И., Шмалько Е.Ю.* Метод многослойного сетевого оператора в задаче синтеза системы управления группой роботов // Труды Второй молодежной научной конференции «Задачи современной информатики». – М.: ФИЦ ИУ РАН. – 2015. – С.241-247.

77. *Дивеев А.И., Шмалько Е.Ю.* Численный синтез системы управления группой роботов // Материалы 8-й Всероссийской мульти-конференции по проблемам управления, Геленджик, Россия. – 2015. – С.168-171.

78. *Дивеев А.И., Шмалько Е.Ю.* Многослойный сетевой оператор для численного синтеза системы управления группой роботов // Фундаментальные проблемы системной безопасности: Материалы школы-семинара молодых ученых, 2-4 июня 2015. – Елец: ЕГУ им. И.А. Бунина. – С.47-52.

79. *Шмалько Е.Ю.* О задаче синтеза системы управления группой роботов // Фундаментальные проблемы системной безопасности: Материалы школы-семинара молодых ученых, 2-4 июня 2015. – Елец: ЕГУ им. И.А.Бунина. – С.148-156.

80. *Дивеев А.И., Шмалько Е.Ю.* Многослойный сетевой оператор в задаче управления группой роботов // Труды VIII международ. научно-практической конференции «Инженерные системы – 2015», 20-22 апреля 2015. – С.172-177.

81. *Дивеев А.И., Шмалько Е.Ю.* Численное решение задачи оптимального управления с фазовыми ограничениями методом вариационного генетического алгоритма // Вопросы теории безопасности и устойчивости систем. – М.: ВЦ РАН, 2014. – Вып.16. – С.99-106

82. *Шмалько Е.Ю.* Синтез управления в задаче координации безопасного движения группой роботов // Фундаментальные проблемы системной безопасности: материалы школы-семинара молодых ученых, 20-22 ноября 2014. – Елец: ЕГУ им. И.А. Бунина, 2014. – С. 123-128.

83. *Дивеев А.И., Шмалько Е.Ю.* Оценка оптимальности численного решения задачи синтеза системы управления // Фундаментальные проблемы системной безопасности: материалы V Международной научной конференции, посвященной 90-летию со дня рождения академика В.Ф. Уткина. – Елец: ЕГУ им. И.А. Бунина, 2014. – С. 478-482.

84. *Дивеев А.И., Шмалько Е.Ю.* Формализация проблемы численного синтеза структур и параметров систем управления // Фундаментальные проблемы системной безопасности: материалы V Международной научной конференции, посвященной 90-летию со дня рождения академика В.Ф. Уткина. – Елец: ЕГУ им. И.А. Бунина, 2014. – С.30-35.

85. *Дивеев А.И., Шмалько Е.Ю.* Численные методы символьной регрессии для решения задачи синтеза управления // Фундаментальные проблемы системной безопасности: материалы V Международной научной конференции, посвященной 90-летию со дня рождения академика В.Ф. Уткина. – Елец: ЕГУ им. И.А. Бунина, 2014. – С.473-478.

86.*Дивеев А.И., Шмалько Е.Ю.* Синтез системы управления группой роботов методом сетевого оператора // Современные проблемы науки и образования. – 2014. – № 4.

87.*Дивеев А.И., Шмалько Е.Ю.* Эволюционный метод синтеза системы управления мобильным роботом // Труды VI международной научно-практической конференции «Инженерные системы – 2013», посвященной столетнему юбилею первого ректора РУДН С.В. Румянцева. – С.127-132.

88.*Дивеев А.И., Шмалько Е.Ю.* Алгоритм интеллектуальной эволюции в задаче синтеза безопасной системы управления мобильным роботом // Вопросы теории безопасности и устойчивости систем. – М.: ВЦ РАН, 2013. – Вып.15. – С.23-35

89.*Дивеев А.И., Шмалько Е.Ю.* Метод символьной регрессии на основе сетевого оператора в задаче синтеза управления // Современные проблемы науки и образования. – 2013. – №3.

90.*Дивеев А.И., Шмалько Е.Ю., Юрков Н.К.* Синтез управления движением мобильного робота по траектории методом интеллектуальной эволюции // Труды Международного симпозиума Надежность и качество 2013. – Пенза: Изд-во ПГУ. 27 мая -03 июня 2013. – С.188-190.

Личный вклад соискателя в получении результатов, изложенных в диссертации. Все основные результаты диссертационной работы получены и обоснованы автором самостоятельно. В работах без соавторов [2, 3, 14, 23, 48, 49, 51, 52] представлены основные положения разработанного принципа синтезированного оптимального управления, изложены теоретические аспекты и рассмотрены методы практической реализации, приведены экспериментальные результаты. В работах [23, 48] отдельное внимание уделено вопросу реализуемости, в [57] приведены результаты исследования вопроса оптимального расположения точек равновесия при синтезированном оптимальном управлении. В [52] показаны возможности применения разработанного подхода для управления мобильным роботом с нестандартной

моделью движения за счет mecanum-колес, в [51] представлена апробация подхода для управления квадрокоптером. В более ранних работах автора [79, 82] рассматривалась задача синтеза управления группой роботов, результаты исследования которой и методы ее решения легли в основу первого этапа предложенного принципа синтезированного оптимального управления, в [68] рассматривались вопросы ускорения работы вычислительных методов синтеза управления, в [73] рассмотрены вопросы возможностей применения методов эволюционных вычислений для решения сложных задач оптимизации.

Большая часть работ автора опубликована в соавторстве со своим научным консультантом и руководителем отдела, в котором уже 13 лет работает автор, профессором Дивеевым А.И. В более ранних работах [9 - 11, 13, 15, 18, 24, 31, 39 - 47, 58, 66 - 70, 72 - 80, 82 - 90], посвященных исследованиям задачи синтеза управления, Шмалько Е.Ю. участвовала в разработке вариационных методов символьной регрессии для решения задач структурно-параметрического синтеза систем управления, предложены способы реализации принципа вариаций базисного решения, разработанного Дивеевым А.И., для различных методов символьной регрессии, таких как вариационное генетическое программирование, вариационное аналитическое программирование, вариационное полное бинарное генетическое программирование, предложен подход многослойной структуры для метода сетевого оператора, разработаны программные модули для реализации предложенных методов, проведены исследования эффективности методов для решения различных задач управления. В работах [36, 70, 81] исследовала возможности применения эволюционных алгоритмов как методов глобальной оптимизации, для решения задачи оптимального управления. Все разрабатываемые методы и алгоритмы, реализованные в виде программных комплексов и применяемые в задачах управления, вошли в единый класс методов машинного обучения управлению, систематично представленный в монографии [1] в соавторстве с руководителем, где личный вклад Шмалько Е.Ю. состоит в концептуализации идеи машинного обучения управлению методами символьной регрессии, в совместной разработке

общих теоретических основ, формулировки задачи синтезированного оптимального управления, представлении вариационных методов символьной регрессии, подготовки экспериментальной части. Данные исследования легли в основу разработки методов решения двух этапов принципа синтезированного оптимального управления [4-8, 10, 13, 15-17, 19 -22, 25 – 30, 32 – 35, 37, 38, 53, 55, 59 - 63]. Предпосылки принципа синтезированного оптимального управления были предложены автором в ранних работах [38, 40, 45], и в дальнейшем был сформулирован сам подход в работах [1-8, 14, 16-20, 22, 23, 25 - 29, 32 - 35, 37, 48, 49, 51-53, 55, 57, 59 - 63], где автору принадлежит концептуализация и формализация метода, математическая постановка задач двух этапов реализации разработанного подхода, предложены и исследованы методы реализации принципа синтезированного оптимального управления, разработаны программные модули и проведены экспериментальные исследования решения задач управления различными робототехническими системами.

Приложение 1. Программный код часто используемых машинно реализуемых функций

В данном приложении приведено описание наиболее часто используемых машинных функций, представленных в виде программного кода на языке Free Pascal, где для функций с одним аргументом используется обозначение Ro_N, для функций с двумя аргументами — Xi_N, а Nu_N используется для функций с тремя аргументами, где N – номер функции.

```
Unit Machine-made functions

const
        infinity=1e8;
        eps=1e-8;
        eps1=1e-2;
        pokmax=8;
        E2c:array [0..7] of real=(0,1,-infinity,infinity,0,0,0,0);

//*******************************
Function Ro_1 (z: real): real;
Begin
        result:=z;
End;

//*******************************
Function Ro_2(z:real):real;
Begin
        if abs(z)>sqrt(infinity)
           then result:=infinity
        else result:=sqr(z);
End;

//*******************************
Function Ro_3(z:real):real;
Begin
        result:=-z;
End;
```

```
//*******************************

Function Ro_4(z:real):real;
Begin
        result:=Ro_10(z)*sqrt(abs(z));
End;

//*******************************
Function Ro_5(z:real):real;
Begin
        if abs(z)>eps
          then result:=1/z
        else result:=Ro_10(z)/eps;
End;

//*******************************
Function Ro_6(z:real):real;
Begin
        if z>-ln(eps)
          then result:=-ln(eps)
        else result:=exp(z);
End;

//*******************************
Function Ro_7(z:real):real;
Begin
        if abs(z)<exp(-pokmax)
          then result:=ln(eps)
        else result:=ln(abs(z));
End;

//*******************************
Function Ro_8(z:real):real;
Begin
        if abs(z)>-ln(eps)
          then result:=Ro_10(z)
        else result:=(1-exp(-z))/(1+exp(-z));
End;

//*******************************
Function Ro_9(z:real):real;
Begin
```

```
        if z>=0
          then result:=1
        else result:=0;
End;

//*******************************
Function Ro_10(z:real):real;
Begin
        if z>=0
          then result:=1
        else result:=-1;
End;

//*******************************
Function Ro_11(z:real):real;
Begin
        result:=cos(z);
End;

//*******************************
Function Ro_12(z:real):real;
Begin
        result:=sin(z);
End;

//*******************************
Function Ro_13(z:real):real;
Begin
        if abs(z)<eps
          then result:=Ro$_-$10(z)*pi/2
        else result:=arctan(z);
End;

//*******************************
Function Ro_14(z:real):real;
Begin
        if abs(z)>Ro_15(infinity)
           then result:=Ro_10(z)*infinity
        else result:=sqr(z)*z;
End;

//*******************************
```

```
Function Ro_15(z:real):real;
Begin
        if abs(z)<eps
          then result:=Ro_10(z)*eps
        else result:=Ro_10(z)*exp(ln(abs(z))/3);
End;

//*******************************
Function Ro_16(z:real):real;
Begin
        if abs(z)<1
          then result:=z
        else result:=Ro_10(z);
End;

//*******************************
Function Ro_17(z:real):real;
Begin
        result:=Ro_10(z)*ln(abs(z)+1);
End;

//*******************************
Function Ro_18(z:real):real;
Begin
        if abs(z)>-ln(eps)
          then result:=Ro_10(z)*infinity
        else result:=Ro_10(z)*(exp(abs(z))-1);
End;

//*******************************
Function Ro_19(z:real):real;
Begin
        if abs(z)>1/eps
          then result:=Ro_10(z)*eps
        else result:=Ro_10(z)*exp(-abs(z));
End;

//*******************************
Function Ro_20(z:real):real;
Begin
        result:=z/2;
End;
```

```
//*******************************
Function Ro_21(z:real):real;
Begin
        result:=2*z;
End;

//*******************************
Function Ro_22(z:real):real;
Begin
        if z<0
          then result:=exp(z)-1
        else Result:=1-exp(-abs(z));
End;

//*******************************
Function Ro_23(z:real):real;
Begin
        if abs(z)>1/eps
          then result:=-Ro_10(z)/eps
        else  result:=z-z*sqr(z);
End;

//*******************************
Function Ro_24(z:real):real;
Begin
        if z>infinity
          then result:=1
        else
          if exp(-z)>infinity
            then result:=0
          else result:=1/(1+exp(-z));
End;

//*******************************
Function Ro_25(z:real):real;
Begin
        if z>0
          then result:=1
        else result:=0;
End;
```

```
//*******************************
Function Ro_26(z:real):real;
Begin
        if abs(z)<eps1
          then result:=0
        else result:=Ro_10(z);
End;

//*******************************
Function Ro_27(z:real):real;
Begin
        if abs(z)>1
          then result:=Ro_10(z)
        else result:=Ro_10(z)*(1-sqrt(1-sqr(z)));
End;

//*******************************
Function Ro_28(z:real):real;
Begin
        if z*z> ln(infinity)
          then result:=z*(1-eps)
        else result:=z*(1-exp(-sqr(z)));
End;

//*******************************
Function Xi_1(z1,z2:real):real;
Begin
        result:=z1+z2;
End;

//*******************************
Function Xi_2(z1,z2:real):real;
Begin
        if abs(z1*z2)> infinity
          then result:=Ro_10(z1*z2)*infinity
        else result:=z1*z2;
End;

//*******************************
Function Xi_3(z1,z2:real):real;
Begin
        if z1>=z2
```

```
          then result:=z1
        else result:=z2;
End;

//*******************************
Function Xi_4(z1,z2:real):real;
Begin
        if z1<z2
          then result:=z1
        else result:=z2;
End;

//*******************************
Function Xi_5(z1,z2:real):real;
Begin
        result:=z1+z2-z1*z2;
End;

//*******************************
Function Xi_6(z1,z2:real):real;
Begin
        result:=Ro_10(z1+z2)*sqrt(sqr(z1)+sqr(z2));
End;

//*******************************
Function Xi_7(z1,z2:real):real;
Begin
        result:=Ro_10(z1+z2)*(abs(z1)+abs(z2));
End;

//*******************************
Function Xi_8(z1,z2:real):real;
Begin
        result:=Ro_10(z1+z2)*Xi_2(abs(z1),abs(z2));
End;

//*******************************
Function Nu_1(z1,z2,z3:real):real;
Begin
        if z1>0
          then result:=z2
        else result:=z3;
```

```
End;

//*******************************
Function Nu_2(z1,z2,z3:real):real;
Begin
        if z1>z2
                then result:=z3
        else result:=-z3;
End;

//*******************************
Function Nu_3(z1,z2,z3:real):real;
Begin
        if z1>0
          then result:=z2+z3
        else result:=z2-z3;
End;

//*******************************
Function Nu_4(z1,z2,z3:real):real;
Begin
        if z1>z2
          then
            if z1>z3
                then result:=z1
            else result:=z3
        else
        if z2>z3
          then result:=z2
        else result:=z3;
End.
```

Приложение 2. Свидетельства автора о государственной регистрации программ для ЭВМ

РОССИЙСКАЯ ФЕДЕРАЦИЯ

СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2012661004

«Метод интеллектуальной эволюции для синтеза систем управления»

Правообладатель(ли): ***Шмалько Елизавета Юрьевна (RU), Дивеев Асхат Ибрагимович (RU)***

Автор(ы): ***Шмалько Елизавета Юрьевна, Дивеев Асхат Ибрагимович (RU)***

Заявка № **2012618446**

Дата поступления **9 октября 2012 г.**

Зарегистрировано в Реестре программ для ЭВМ

5 декабря 2012 г.

Руководитель Федеральной службы по интеллектуальной собственности

Б.П. Симонов

РОССИЙСКАЯ ФЕДЕРАЦИЯ

СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2014615467

«Метод вариационного аналитического программирования для синтеза системы управления»

Правообладатели: ***Дивеев Асхат Ибрагимович (RU), Шмалько Елизавета Юрьевна (RU)***

Авторы: ***Дивеев Асхат Ибрагимович (RU), Шмалько Елизавета Юрьевна (RU)***

Заявка № **2014613325**
Дата поступления **10 апреля 2014 г.**
Дата государственной регистрации
в Реестре программ для ЭВМ ***27 мая 2014 г.***

Руководитель Федеральной службы по интеллектуальной собственности

Б.П. Симонов

РОССИЙСКАЯ ФЕДЕРАЦИЯ

СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2014617621

«Программный комплекс для решения задачи синтеза управления методом вариационного генетического программирования»

Правообладатели: ***Дивеев Асхат Ибрагимович (RU), Шмалько Елизавета Юрьевна (RU)***

Авторы: ***Дивеев Асхат Ибрагимович (RU), Шмалько Елизавета Юрьевна (RU)***

Заявка № **2014613287**
Дата поступления **10 апреля 2014 г.**
Дата государственной регистрации
в Реестре программ для ЭВМ ***29 июля 2014 г.***

Руководитель Федеральной службы по интеллектуальной собственности

Б.П. Симонов

РОССИЙСКАЯ ФЕДЕРАЦИЯ

СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2014610171

«Вариационный генетический алгоритм для решения задачи оптимального управления»

Правообладатели: ***Дивеев Асхат Ибрагимович (RU), Шмалько Елизавета Юрьевна (RU)***

Авторы: ***Дивеев Асхат Ибрагимович (RU), Шмалько Елизавета Юрьевна (RU)***

Заявка № **2013660161**

Дата поступления **06 ноября 2013 г.**

Дата государственной регистрации в Реестре программ для ЭВМ ***09 января 2014 г.***

Руководитель Федеральной службы по интеллектуальной собственности

Б.П. Симонов

РОССИЙСКАЯ ФЕДЕРАЦИЯ

СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2015617366

«Интерпретатор программной записи математического выражения в матрицу сетевого оператора»

Правообладатели: ***Дивеев Асхат Ибрагимович (RU), Шмалько Елизавета Юрьевна (RU)***

Авторы: ***Дивеев Асхат Ибрагимович (RU), Шмалько Елизавета Юрьевна (RU)***

Заявка № **2015614429**
Дата поступления **21 мая 2015 г.**
Дата государственной регистрации
в Реестре программ для ЭВМ ***08 июля 2015 г.***

Врио руководителя Федеральной службы по интеллектуальной собственности

Л.Л. Кирий

РОССИЙСКАЯ ФЕДЕРАЦИЯ

СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2015617591

«Программа расчета математического выражения с помощью многослойного сетевого оператора»

Правообладатели: ***Дивеев Асхат Ибрагимович (RU), Шмалько Елизавета Юрьевна (RU)***

Авторы: ***Дивеев Асхат Ибрагимович (RU), Шмалько Елизавета Юрьевна (RU)***

Заявка № **2015614591**
Дата поступления **21 мая 2015 г.**
Дата государственной регистрации
в Реестре программ для ЭВМ ***15 июля 2015 г.***

Врио руководителя Федеральной службы по интеллектуальной собственности

Л.Л. Кирий

РОССИЙСКАЯ ФЕДЕРАЦИЯ

СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2016619672

«Программный комплекс для решения задачи стабилизации методом многослойного сетевого оператора»

Правообладатели: ***Дивеев Асхат Ибрагимович (RU), Шмалько Елизавета Юрьевна (RU)***

Авторы: ***Дивеев Асхат Ибрагимович (RU), Шмалько Елизавета Юрьевна (RU)***

Заявка № **2016617014**
Дата поступления **30 июня 2016 г.**
Дата государственной регистрации
в Реестре программ для ЭВМ ***25 августа 2016 г.***

Руководитель Федеральной службы по интеллектуальной собственности

Г.П. Ивлиев

РОССИЙСКАЯ ФЕДЕРАЦИЯ

СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2017662486

«Программный комплекс для решения задачи стабилизации роботов методом бинарного вариационного генетического программирования»

Правообладатели: ***Шмалько Елизавета Юрьевна (RU), Дивеев Асхат Ибрагимович (RU)***

Авторы: ***Шмалько Елизавета Юрьевна (RU), Дивеев Асхат Ибрагимович (RU)***

Заявка № **2017618933**
Дата поступления **22 августа 2017 г.**
Дата государственной регистрации
в Реестре программ для ЭВМ ***09 ноября 2017 г.***

Руководитель Федеральной службы по интеллектуальной собственности

Г.П. Ивлиев

РОССИЙСКАЯ ФЕДЕРАЦИЯ

СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2017662485

«Программа автоматического синтеза генератора оптимальных траекторий для группы роботов методом многослойного сетевого оператора»

Правообладатели: ***Шмалько Елизавета Юрьевна (RU), Дивеев Асхат Ибрагимович (RU)***

Авторы: ***Шмалько Елизавета Юрьевна (RU), Дивеев Асхат Ибрагимович (RU)***

Заявка № **2017618934**

Дата поступления **22 августа 2017 г.**

Дата государственной регистрации в Реестре программ для ЭВМ ***09 ноября 2017 г.***

Руководитель Федеральной службы по интеллектуальной собственности

Г.П. Ивлиев

РОССИЙСКАЯ ФЕДЕРАЦИЯ

СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2021611899

«Программа для синтеза системы стабилизации на основе Вариационного Декартового Генетического Программирования»

Правообладатель: ***Федеральное государственное учреждение «Федеральный исследовательский центр «Информатика и управление» Российской академии наук» (ФИЦ ИУ РАН) (RU)***

Авторы: ***Дивеев Асхат Ибрагимович (RU), Шмалько Елизавета Юрьевна (RU)***

Заявка № **2021610671**

Дата поступления **25 января 2021 г.**

Дата государственной регистрации в Реестре программ для ЭВМ ***08 февраля 2021 г.***

Руководитель Федеральной службы по интеллектуальной собственности

Г.П. Ивлиев

РОССИЙСКАЯ ФЕДЕРАЦИЯ

СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2022664577

«Программа расчета синтезированного оптимального управления методом роя частиц для объекта управления с системой стабилизации»

Правообладатель: ***Федеральное государственное учреждение «Федеральный исследовательский центр «Информатика и управление» Российской академии наук» (ФИЦ ИУ РАН) (RU)***

Авторы: ***Шмалько Елизавета Юрьевна (RU), Дивеев Асхат Ибрагимович (RU)***

Заявка № **2022663829**
Дата поступления **21 июля 2022 г.**
Дата государственной регистрации
в Реестре программ для ЭВМ ***02 августа 2022 г.***

Руководитель Федеральной службы
по интеллектуальной собственности

Ю.С. Зубов

РОССИЙСКАЯ ФЕДЕРАЦИЯ

СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2022681057

«Программа расчета синтезированного оптимального управления генетическим алгоритмом оптимизации»

Правообладатель: ***Федеральное государственное учреждение «Федеральный исследовательский центр «Информатика и управление» Российской академии наук» (ФИЦ ИУ РАН) (RU)***

Авторы: ***Шмалько Елизавета Юрьевна (RU), Дивеев Асхат Ибрагимович (RU)***

Заявка № **2022680295**
Дата поступления **31 октября 2022 г.**
Дата государственной регистрации
в Реестре программ для ЭВМ ***09 ноября 2022 г.***

Руководитель Федеральной службы по интеллектуальной собственности

Ю.С. Зубов

РОССИЙСКАЯ ФЕДЕРАЦИЯ

СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2022681183

«Программа расчета синтезированного оптимального управления популяционным алгоритмом оптимизации серых волков»

Правообладатель: ***Федеральное государственное учреждение «Федеральный исследовательский центр «Информатика и управление» Российской академии наук» (ФИЦ ИУ РАН) (RU)***

Авторы: ***Шмалько Елизавета Юрьевна (RU), Дивеев Асхат Ибрагимович (RU)***

Заявка № **2022680300**
Дата поступления **31 октября 2022 г.**
Дата государственной регистрации
в Реестре программ для ЭВМ ***10 ноября 2022 г.***

Руководитель Федеральной службы
по интеллектуальной собственности

Ю.С. Зубов

РОССИЙСКАЯ ФЕДЕРАЦИЯ

СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2022681267

«Программа расчета синтезированного оптимального управления для группы нелинейных динамических объектов»

Правообладатель: ***Федеральное государственное учреждение «Федеральный исследовательский центр «Информатика и управление» Российской академии наук» (ФИЦ ИУ РАН) (RU)***

Авторы: ***Шмалько Елизавета Юрьевна (RU), Дивеев Асхат Ибрагимович (RU)***

Заявка № **2022680297**
Дата поступления **31 октября 2022 г.**
Дата государственной регистрации
в Реестре программ для ЭВМ ***10 ноября 2022 г.***

Руководитель Федеральной службы
по интеллектуальной собственности

Ю.С. Зубов

РОССИЙСКАЯ ФЕДЕРАЦИЯ

СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2022681059

«Программа расчета синтезированного оптимального управления для задач с неопределенностями начальных условий»

Правообладатель: ***Федеральное государственное учреждение «Федеральный исследовательский центр «Информатика и управление» Российской академии наук» (ФИЦ ИУ РАН) (RU)***

Авторы: ***Шмалько Елизавета Юрьевна (RU), Дивеев Асхат Ибрагимович (RU)***

Заявка № **2022680292**
Дата поступления **31 октября 2022 г.**
Дата государственной регистрации
в Реестре программ для ЭВМ ***09 ноября 2022 г.***

Руководитель Федеральной службы
по интеллектуальной собственности

Ю.С. Зубов

Приложение 3. Акты о внедрении результатов диссертационной работы

Инжиниринговый центр
«Автоматика и робототехника»
МГТУ им. Н.Э. Баумана

Россия, 105005, г. Москва, вн. тер. г. муниципальный округ Басманный, ул. Бауманская, д. 57Б, помещ. 1/1
Телефон/факс 8 (499) 263-61-90

Общество с ограниченной ответственностью «Инжиниринговый центр «Автоматика и робототехника МГТУ им. Н.Э. Баумана»
ИНН 9701125029 КПП 770101001 ОГРН 5187746002428

УТВЕРЖДАЮ
Директор
Д.А. Астафьев

«26» июля 2023

АКТ

о внедрении результатов диссертационной работы

Шмалько Елизаветы Юрьевны

на тему: «Принцип синтезированного оптимального управления в робототехнических системах», представленной на соискание ученой степени доктора технических наук

Комиссия в составе:

- председателя: Начальник отдела специальных транспортных систем, к.т.н. К.Е. Бяков;
- членов: инженер ОСТС В.Б. Холоденко, инженер НПО В.Н. Казьмин

составили настоящий акт о том, что результаты, полученные в диссертационной работе Шмалько Е.Ю., а именно:

1) метод решения задачи оптимального управления на основе принципа синтезированного оптимального управления,
2) синтез системы стабилизации на основе машинного обучения методом символьной регрессии,
3) система управления робота на основе оптимального расположения точек равновесия,

использованы при выполнении НИОКР по теме: «Исследования по внедрению робототехнических и автоматизированных решений в технологию выполнения работ в труднодоступных районах с применением мобильных комплексов на базе гусеничных снегоболотоходов» в части реализации системы управления автономным движением машины.

Председатель комиссии: К.Е. Бяков

Члены комиссии: В.Б. Холоденко

В.Н. Казьмин

УТВЕРЖДАЮ
Первый заместитель
генерального директора
ФАУ «ЦАГИ»

Медведский А.Л.
«27» 11 2023 г.

АКТ

о внедрении результатов диссертационной работы
Шмалько Елизаветы Юрьевны

на тему: «Принцип синтезированного оптимального управления в робототехнических системах», представленной на соискание ученой степени доктора технических наук.

Комиссия в составе:

председателя:
- заместителя генерального директора по ВВиСТ, Гранича В.Ю., к.т.н.;

членов:
- начальника сектора 40 НИО-15, Яновой О.В., к.т.н.;
- старшего научного сотрудника НИО-15, Голикова А.А.

составили настоящий акт о том, что результаты, полученные в диссертационной работе Шмалько Е.Ю., а именно:

1) метод решения задачи оптимального управления на основе принципа синтезированного оптимального управления;
2) синтез системы стабилизации БЛА мультироторного типа, выполненный на основе машинного обучения с использованием метода символьной регрессии;
3) ключевые особенности методов, применяемых для оптимизации управления квадрокоптером, на основе принципа синтезированного оптимального управления;
4) эволюционные алгоритмы решения задач глобальной оптимизации, реализующие параллельный поиск решений одновременно в нескольких областях

использованы при выполнении научно-исследовательской работы по теме «Разработка математических моделей и алгоритмов системы управления беспилотных летательных аппаратов вертикального взлета и посадки с циклическими движителями с целью обеспечения управляемости и устойчивости» в части исследования возможностей предложенного алгоритма управления в САУ БЛА мультироторного типа (квадрокоптера) для обеспечения устойчивого автономного движения БЛА в условиях случайных помех и нестационарных ветровых возмущений при движении вблизи объектов в приземном пограничном слое.

Председатель комиссии: _____ Гранич В.Ю.

Члены комиссии: _____ Янова О.В.

_____ Голиков А.А.

Общество с ограниченной ответственностью
«Научно-производственное объединение НаукаСофт»

129085, г. Москва, ул. Годовикова, д. 9, стр. 4, этаж 1, пом/ком 1.1/1.1.4
+7 (495) 255-36-35
contacts@naukasoft.ru
http://naukasoft.ru

УТВЕРЖДАЮ
Исполнительный директор
ООО «НПО НаукаСофт»
_____ А.В. Парфенов
«29» 09 2023 г.

АКТ
о внедрении результатов диссертационной работы Шмалько Елизаветы Юрьевны на тему «Принцип синтезированного оптимального управления в робототехнических системах», представленной на соискание ученой степени доктора технических наук

Комиссия в составе:
председателя – Жмурова Б.В., главного конструктора-заместителя генерального директора, доцента, к.т.н.,
и членов:
Давидова А.О., начальника научно-исследовательского отдела, д.т.н.,
Иванова А.В., начальника конструкторско-технологического отдела, к.т.н.
составила настоящий акт о том, что результаты, полученные в диссертационной работе Шмалько Е.Ю., а именно:

1) метод решения задачи оптимального управления на основе принципа синтезированного оптимального управления;
2) синтез системы стабилизации на основе машинного обучения методом символьной регрессии;
3) система управления квадрокоптера на основе принципа синтезированного оптимального управления

использованы при выполнении опытно-конструкторской работы по теме «Разработка демонстратора радиолокационного комплекса инженерной разведки», шифр «Археолог» в части реализации системы управления автономным движением объекта.

Председатель комиссии: _____ Б.В. Жмуров

Члены комиссии: _____ А.О. Давидов

_____ А.В. Иванов

Корпорация «Тактическое ракетное вооружение»

АКЦИОНЕРНОЕ ОБЩЕСТВО
«ВОЕННО-ПРОМЫШЛЕННАЯ КОРПОРАЦИЯ
«НАУЧНО-ПРОИЗВОДСТВЕННОЕ
ОБЪЕДИНЕНИЕ МАШИНОСТРОЕНИЯ»
(АО «ВПК «НПО машиностроения»)
ул. Гагарина, д. 33, г. Реутов, Московская область, 143966
телеграфный: Реутов Московской ВЕСНА (АТ346416)
Тел.: (495) 528-30-18 (канцелярия) Факс: (495) 302-20-01
E-mail: vpk@npomash.ru http://www.npomash.ru
ОКПО 07501739, ОГРН 1075012001492
ИНН/КПП 5012039795/509950001

Ученый секретарь НТС
АО «ВПК «НПО машиностроения»,
кандидат физ. мат. наук
Л.С.Точилов
2023г.

Исх. № 14/НТК
на № от

АКТ

о внедрении результатов диссертационной работы Шмалько Елизаветы Юрьевны, выполненной на тему: «Принцип синтезированного оптимального управления в робототехнических системах», представленной на соискание ученой степени доктора технических наук.

Комиссия в составе:

председатель комиссии – советник по науке, д.т.н. Яковлев О.В.,
члены комиссии – руководитель проектной дирекции Харламов И.В.,
– главный научный сотрудник, к.т.н. Петровский В.С.,

составила настоящей акт о том, что материалы указанной диссертационной работы были использованы при проведении критически важного ряда исследований: по облику перспективных аэрокосмических беспилотных аппаратов и систем, повышению их специальных целевых возможностей, решению задач эффективного функционирования, специфики конструирования, реализуемости перечисленных инноваций. Использованные разработки включены в материалы отчетов по НИР (инв. 054385, 054713), выполняемых в рамках инновационного АО ВПК «НПО машиностроения».

В проводимых НИР были использованы следующие материалы диссертации:

- задачи и методы оптимального управления робототехническими системами;
- алгоритм реализации принципа синтезированного оптимального управления.
- о решении проблем разработки и реализации автоматизированных методов создания систем управления робототехническими системами;
- результаты моделирования движения различных робототехнических объектов.

Председатель комиссии Яковлев О.В.
Члены комиссии Харламов И.В.
Петровский В.С.