

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)»

На правах рукописи



Конырбаев Нурбек Беркинбайулы

**РЕШЕНИЕ ЗАДАЧИ СИНТЕЗА СИСТЕМЫ УПРАВЛЕНИЯ
ПРОСТРАНСТВЕННЫМ ДВИЖЕНИЕМ ЛЕТАЮЩЕГО РОБОТА
МЕТОДОМ ВАРИАЦИОННОГО АНАЛИТИЧЕСКОГО
ПРОГРАММИРОВАНИЯ**

Специальность 05.07.09 –

Динамика, баллистика, управление движением летательных аппаратов

Диссертация на соискание ученой степени

кандидата технических наук

Научный руководитель:
доктор технических наук, профессор
Разумный Ю.Н.

Москва-2018

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	3
1. ЗАДАЧА СИНТЕЗА УПРАВЛЕНИЯ ГРУППОЙ КВАДРОКОПТЕРОВ И ЕЕ РЕШЕНИЕ МЕТОДОМ ВАРИАЦИОННОГО АНАЛИТИЧЕСКОГО ПРОГРАММИРОВАНИЯ.....	20
1.1. Формальная постановка задачи.....	23
1.2. Задача поиска оптимальных пространственных траекторий	24
1.3. Задача синтеза управления.....	25
1.4. Математическая модель квадрокоптера	29
1.5. Методы решения задачи поиска оптимальных траекторий	33
1.6. Обзор численных методов решения задачи синтеза управления	34
1.7. Вариационный генетический алгоритм для решения задачи поиска оптимальных траекторий.....	51
1.8. Выводы к главе 1	57
2. МЕТОД ВАРИАЦИОННОГО АНАЛИТИЧЕСКОГО ПРОГРАММИРОВАНИЯ ДЛЯ СИНТЕЗА СИСТЕМЫ ОПТИМАЛЬНОГО УПРАВЛЕНИЯ	59
2.1. Вариационное аналитическое программирование	60
2.2. Генетический алгоритм на основе малых вариаций базисного решения... ..	70
2.3. Синтез системы управления методом вариационного аналитического программирования	73
2.4. Выводы к главе 2.....	78
3. СИНТЕЗ СИСТЕМЫ УПРАВЛЕНИЯ ЛЕТАЮЩИМ РОБОТОМ	79
3.1. Задача синтеза оптимального управления летающим роботом	80
3.2. Вычислительный эксперимент	83
3.3. Выводы к главе 3	97
ЗАКЛЮЧЕНИЕ	98
СПИСОК ЛИТЕРАТУРЫ	100

ВВЕДЕНИЕ

В работе рассматривается задача синтеза системы автоматического управления летающим роботом, квадрокоптером. Областью исследования диссертационной работы являются методы и алгоритмы структурно-параметрического синтеза систем автоматического управления сложными динамическими объектами.

В диссертационной работе рассматриваются прикладная задача мониторинга области с помощью группы квадрокоптеров. В задаче заданы точки в трехмерном геометрическом пространстве некоторой определенной области. Необходимо с помощью группы квадрокоптеров пройти все заданные точки. Считаем, что все квадрокоптеры стартуют из одной точки. В каждую из заданных точек должен попасть хотя бы один квадрокоптер. Все квадрокоптеры должны вернуться в начальную точку старта.

Задачу решаем в два этапа. На первом этапе разбиваем все множество заданных точек на подмножества. Количество подмножеств равно количеству квадрокоптеров. Каждое из подмножеств представляет собой набор точек для определенного квадрокоптера. Определяем траектории движения квадрокоптеров с помощью установления порядка прохождения точек каждым квадрокоптером. Упорядочивание точек в подмножествах выполняем по критерию минимальной длины суммарного пути всех квадрокоптеров.

На втором этапе решаем задачу синтеза системы управления квадрокоптером. Система управления включает систему стабилизации квадрокоптера относительно заданной точки в пространстве состояний квадрокоптера. В данном случае считаем, что геометрическое трехмерное пространство является подпространством пространства состояний квадрокоптера, в которое вместе с геометрическими координатами центра масс квадрокоптера входят координаты углов, угловых и линейных скоростей квадрокоптера. Система стабилизации квадрокоптера должна обеспечить устойчивость квадрокоптера относительно заданной точки в пространстве состояний. В систему управления

квадрокоптером вместе с системой стабилизации входит блок переключения точек, который осуществляет при достижении квадрокоптером окрестности заданной точки пространства состояний переключение текущей точки на следующую точку геометрического трехмерного подпространства состояний.

Задача первого этапа, нахождения траекторий движения квадрокоптеров, представляет собой расширение известной задачи коммивояжера к задаче поиска путей для группы коммивояжеров. При решении задачи в работе применяем эволюционный генетический алгоритм. Задача коммивояжера относится к классу NP-полных задач, для которых не известны алгоритмы решения с полиномиальной скоростью сходимости.

При решении задачи второго этапа, задачи синтеза управления используем новый метод символьной регрессии, метод вариационного аналитического программирования. Метод представляет собой поиск закодированного специальным образом математического выражения, описывающего функциональную зависимость управления от координат пространства состояний, с помощью эволюционного генетического алгоритма. Разработка нового метода для решения задачи синтеза управления является основным теоретическим результатом работы. Задача синтеза управления является существенно более сложной и практически важнее, чем NP-полная задача коммивояжера.

Решение задачи синтеза управления является сложной проблемой, так как требует нахождения математического выражения, описывающего функциональную зависимость управления от координат пространства состояний объекта. Определяемые по искомому математическому выражению значения вектора управления должны обеспечивать достижение объектом заданной цели управления с оптимальным значением заданного критерия качества. Искомую функцию называем в дальнейшем синтезирующей функцией, и она должна обеспечивать достижение заданной цели управления с оптимальным значением критерия качества для любого начального состояния из некоторой заданной области пространства состояний объекта.

Настоящая диссертационная работа посвящена численному решению задачи синтеза управления методом вариационного аналитического программирования. В качестве объекта управления рассматривается группа мобильных летающих роботов типа квадрокоптеров, решающих задачу мониторинга заданного района.

Задача синтеза управления является важнейшей задачей в области управления, не имеющей в общем случае методов решения. На этапе создания автоматических систем управления, работающих без участия человека, разработчик сталкивается с проблемой решения задачи синтеза управления. Для всех разрабатываемых систем автоматического управления требуется создать блок управления, в котором по состоянию объекта управления и заданной цели управления необходимо вырабатывать управляющие сигналы, обеспечивающие достижение объектом цели управления, с некоторым оптимальным значением критерия качества. Сегодня в подавляющем большинстве случаев данная задача решается «вручную» на основе интуиции, опыта и средств моделирования систем управления. Такой метод решения задачи синтеза управления называется техническим синтезом.

Несмотря на существующую математическую формулировку задачи синтеза управления, как задачи оптимизационной, требующей нахождения управления, обеспечивающего оптимальное значение некоторого критерия качества, технический метод синтеза опирается в основном на модели систем управления аналогичных устройств без учета критерия оптимальности. Далее на основе анализа результатов моделирования разрабатываемой системы управления принимаются, в основном, субъективные решения по улучшению качества управления и изменению структуры блока управления. Оптимизационные задачи в процессе разработки систем управления решаются для уточнения значений параметров спроектированной системы управления, т.е. в автоматическом режиме решается только задача параметрического синтеза управления.

Вычислительные методы сегодня позволяют практически моделировать любую систему управления. Оптимизационные алгоритмы, хотя не всегда тривиально, способны определять оптимальные значения параметров любых

сложных систем управления. Структурный синтез систем управления в подавляющем большинстве случаев делается «вручную» без участия вычислительных оптимизационных процедур. Такое состояние в разработке систем автоматического управления сильно тормозит их развитие и не позволяет находить решения для сложных задач управления, которые сегодня возникают в связи с бурным развитием робототехнических устройств. Новейшие методы символьной регрессии с помощью специальных видов кодирования математических выражений и эволюционных алгоритмов позволяют сегодня решать численно задачу структурного синтеза. Диссертационная работа посвящена применению одного из методов символьной регрессии, метода вариационного аналитического программирования, для решения задачи синтеза управления. Тема диссертационной работы посвящена исследованию и разработке метода решения востребованной, но не решенной сегодня задачи синтеза системы автоматического управления, что определяет актуальность темы диссертационных исследований.

В диссертационной работе разработан и исследован метод вариационного аналитического программирования для решения задачи синтеза управления. Метод реализован в виде комплекс программ и исследован на примерах решения задачи синтеза управления. Степень разработанности диссертационных исследований доведена до практической реализации в виде программного продукта.

Возможность создания вычислительных методов и алгоритмов для поиска структур математических выражений появилась с созданием методов символьной регрессии. Ранее структуры математических выражений задавались исследователями с точностью до значений параметров. Далее с помощью какого-либо вычислительного, как правило оптимизационного, алгоритма определяли оптимальные значения параметров, удовлетворяющих условиям задачи, в большинстве обеспечивающих оптимальное значение некоторому заданному критерию качества. К этому классу методов поиска функций относятся методы, использующие всевозможные полиномы, ряды, вейвлет-функции и т.п. К данному

классу методов следует также отнести и метод нейронных сетей. Здесь структура сети выбирается исследователем, иногда с точностью до числа слоев. Обучение сети представляет собой поиск оптимальных значений большого числа весовых коэффициентов. В некоторых случаях при обучении в сети изменяется число слоев, т. е. изменяется структура сети, но изменение структуры нейронной сети осуществляется регулярно, приблизительно как изменение членов ряда.

Методы символьной регрессии существенно отличаются от ранее известных методов поиска функций. Методы символьной регрессии включают метод кодирования функции, как правило код функции – это набор целых чисел, описывающих композицию вложенных друг в друга элементарных функций, и алгоритм поиска оптимальной структуры функции на множестве возможных кодов. Общепринято, что методы символьной регрессии появились с созданием метода генетического программирования в 1992 году. Автор метода профессор, университета Стэнфорда Дж. Коза решал задачу автоматического написания программ и искал тексты программ в виде универсального кода префиксной символьной записи. Он применил известный к тому времени генетический алгоритм к поиску кода программы. Естественно, что множество кодов программ шире множества кодов математических выражений, так как любое математическое выражение, особенно то, которое используется при синтезе управления объектами, может быть вычислено с помощью компьютерной программы, поэтому создание метода генетического программирования является новым этапом развитие методов поиска структуры математического выражения.

Для решения задачи синтеза управления в работе используется численный метод вариационного аналитического программирования. Метод аналитического программирования [65, 83–87] разработал профессор И. Зелинка в 2002 году. Метод аналитического программирования представляет собой развитие метода генетического программирования Дж. Козы [76].

Из известных методов символьной регрессии: метода генетического программирования [76–78], метода грамматической эволюции [81, 82], метода аналитического программирования [65, 83–87] и метода сетевого оператора [15–

18]), которые позволяют находить структуры математических выражений в закодированной форме с помощью эволюционных алгоритмов. Метод аналитического программирования имеет самый компактный код в виде набора чисел, указывающих на относительные номера функций из некоторого заданного упорядоченного множества функций. Приведем краткие характеристики методов символьной регрессии.

Метод генетического программирования кодирует математическое выражение в форме префиксной записи символов, описывающих операции и операнды математического выражения. Каждый символ записи соответствует некоторой операции, а в случае поиска математического выражения, некоторой функции из заданной таблицы функций. По таблице определяют вид функции и количество ее аргументов. Следующие символы являются аргументами функции, на которую указывал предыдущий символ. Аргументами функции могут быть переменные, параметры или другие функции числовые параметры и переменные, которые входят в описание математического выражения, также размещены в таблице функций и определены как нульместные функции или функции без аргументов. Цепочка символов, которая описывает один аргумент всегда заканчивается символом нульместной функции. После символа-кода нульместной функции начинается описание следующего аргумента функции.

Поиск искомого математического выражения в методе генетического программирования осуществляют с помощью генетического алгоритма, который выполняет генетические операции на строках символьных записей. При выполнении операции скрещивания в генетическом программировании находят в каждой из строк символов точки скрещивания. Каждая из точек соответствует символу функции, причем функции в обеих скрещиваемых строках символов должны иметь одинаковые количества аргументов. Если точки скрещивания соответствуют указанным требованиям, то в каждой из строк символов находят подстроки, которые соответствуют самостоятельным математическим выражениям. Далее найденные подстроки в строках символов обмениваются, т.е. из каждой строки удаляется найденная подстрока и вставляется в точку

скрещивания другой строки символов. В результате скрещивания получаются новые математические выражения. Операция мутации в генетическом программировании либо не делается, либо выполняется заменой номера функции, другим случайно сгенерированным номером функции с тем же количеством аргументов. Выполнение генетической операции скрещивания требует существенных вычислительных затрат, часто может не выполняться из-за несоответствия установленным требованиям найденных точек скрещивания и приводит к изменению длины кода. Эти обстоятельства послужили причиной усовершенствования генетического программирования и созданию новых методов.

Метод грамматической эволюции [81, 82] предназначен для поиска кода программы в универсальной форме Бэкуса – Наура. Коды элементов формы называются кодонами и записываются в двоичной форме строкой из восьми бит. Операции скрещивания выполняются на битовых строках обычным для генетического алгоритма способом, обменом участков битовых строк «хвостов» после точки скрещивания. Операция мутации выполняется также классическим для генетического алгоритма методом, случайной заменой в строке кода в случайно выбранной позиции бита. В результате классических операций скрещивания и мутации коды универсальной формы записей программ или математических выражений могут быть нарушены. Для исправления ошибок кодов вводится дополнительная операция коррекции, которая заключается либо в исключении лишних кодов, либо в дополнении кодов терминальных символов. В случае поиска математических выражений терминальные символы формы Бэкуса - Наура соответствуют нульместным функциям.

Метод аналитического программирования [65, 83–87] при решении задачи поиска математического выражения существенно не отличается от метода грамматической эволюции. Все элементы математического выражения, т.е. функции в том числе и нульместные функции, которые соответствуют кодам переменных и параметров, объединены в одно упорядоченное множество функций. Номер элемента в данном объединенном множестве соответствует его

коду, причем по неизвестной причине в аналитическом программировании используется относительная нумерация кодирования функций, т.е. номер функции вычисляется относительно номера предыдущего элемента. Первая функция соответствует функции под номером, указываемым первым числом в объединенном множестве. Вторая функция соответствует номеру, равному сумме первого и второго чисел. Третья функция соответствует номеру равному сумме всех трех чисел, и т.д. Если в объединенном множестве количество элементов меньше, чем вычисленный код функции, то для вычисления номера функции берем остаток от деления вычисленного кода на количество элементов в объединенном множестве. Существенный смысл такой формы кодирования авторами аналитического программирования не объясняется, поэтому никакой разницы между указанием в виде кода абсолютного номера функции в объединенном множестве или числа, указывающего на номер функции относительно номера функции кода предыдущего элемента нет, кроме дополнительных операций вычисления абсолютного номера функции.

Метод аналитического программирования имеет самую компактную форму кодирования математического выражения. При поиске решения с помощью генетического алгоритма после выполнения операций скрещивания, которая осуществляется обменом оставшихся после точки скрещивания кодов функций, и мутации, которая выполняется случайной заменой кода в случайно выбранной позиции, также возникает необходимость коррекции кода записи математического выражения, которая решается либо отбрасыванием последних элементов записи, либо добавлением в конец записи кодов номеров из подмножества нульместных функций. В методе аналитического программирования для поиска решения используется не только генетический алгоритм, но и другие эволюционные алгоритмы, метод дифференциальной эволюции и метод роя частиц и т.п.

Все три перечисленных метода символьной регрессии: метод генетического программирования, метод грамматической эволюции и метод аналитического программирования, первоначально не предназначались для решения задачи

синтеза управления. Особенностью задачи синтеза управления [1, 3, 11, 49, 58] является то, что помимо оптимального значения критерия, по которому отбирается возможное решение, каждое решение должно также удовлетворять некоторой краевой задаче, которая в терминах теории управления трактуется, как цель управления. В большинстве случаев случайно сгенерированное закодированное возможное решение не удовлетворяет условиям достижения цели. Для сложных задач управления только очень небольшой процент из случайно сгенерированного множества возможных решений удовлетворяет требованию достижения цели управления, поэтому операциям эволюционного алгоритма поиска необходимо подвергать только малую часть начального множества возможных решений. Из-за особенности задачи синтеза известные методы символьной регрессии довольно редко применяются для решения задачи синтеза управления и требуют при поиске решения больших вычислительных ресурсов для работы с большими массивами.

Метод сетевого оператора [15–18, 69–71] разрабатывался как численный метод символьной регрессии для решения задачи синтеза управления, поэтому в нем предусматривается возможность поиска решения на множестве функций, так или иначе удовлетворяющих условиям достижения цели управления. Основным инструментом, обеспечивающим возможность эффективно решать задачи синтеза управления является применение при поиске принципа малых вариаций базисного решения. Суть принципа заключается в том, что исследователь задает одно базисное решение в виде кода, определяет все возможные малые вариации этого кода, которые приводят к другому коду возможного решения, определяет форму кодирования малых вариаций, и осуществляет поиск решения на множестве кодов малых вариаций базисного решения. Данный подход является существенным для решения задачи синтеза управления, поскольку любой специалист в области управления всегда может задать базисное решение, например, в виде линейной отрицательной обратной связи по отклонению от терминального состояния, и искать с помощью вычислительного алгоритма решение, которое оказывается лучше базисного, сэкономив, тем самым, огромное количество времени на поиск

какого-то решения, которое удовлетворяет терминальным условиям. Метод сетевого оператора показал свою эффективность при решении различных задач управления. Существенным ограничением метода сетевого оператора является ограниченная возможность поиска решения, в виде математического выражения, построенного только на функциях с одним или двумя аргументами. Для синтеза сложных логико-функциональных систем управления метод искал решения в виде двух сетевых операторов различного типа, один сетевой оператор описывал выражения, состоящие из арифметических функций, а второй из функций многозначной логики.

Попытка применить принцип малых вариаций базисного решения, заимствованный из метода сетевого оператора, для методов генетического программирования, грамматической эволюции и аналитического программирования с целью их наиболее эффективного применения к решению задачи синтеза управления привела к появлению новых методов символьной регрессии: метода вариационного генетического программирования [32], метода вариационной грамматической эволюции и метода вариационного аналитического программирования [19, 23, 67, 72]. При создании новых вариационных методов символьной регрессии модифицируется или уточняется код записи математического выражения с целью более конкретного определения показателей вариации и определяются малые вариации кода.

Общая идея определения малой вариации кода – это удовлетворение требованиям полноты и малости. Свойства полноты набора малых вариаций определяется тем, чтобы с помощью конечного набора этих вариаций всегда возможно из кода одного возможного решения получить код любого другого возможного решения. Малость вариации это свойства, которое указывает на то, что малая вариация не может быть разбита на более мелкие вариации.

Принцип малых вариаций базисного решения позволяет при решении задачи синтеза управления задать базисное решение, которое в общем виде позволяет достичь объектом цели управления. Все малые вариации такого базисного решения будут опираться на решение, обеспечивающее достижение

цели управления, и модифицировать его с помощью малых вариаций в направлении улучшения показателей качества.

Целью применения метода вариационного аналитического программирования для синтеза системы управления вместо метода сетевого оператора является расширение множества функций, из которых формируется математическое выражение, описывающее зависимость управления от координат состояния объекта, в частности использование при синтезе функций с тремя и больше аргументами. К таким функциям относятся функции, которые могут выполнять действия, аналогичные условному оператору «if». Наличие оператора условия довольно часто требуется в современных интеллектуальных системах управления. При создании системы управления движением летающего робота [4–7] по пространственной траектории оператор условия необходим, чтобы осуществлять переключение целевой точки управления. Заметим, что решение подобной задачи синтеза с помощью метода сетевого оператора привело к созданию логико-функциональной системы управления реализация которой потребовала нахождения двух сетевых операторов, арифметического сетевого оператора, описывающего математическое выражение системы пространственной стабилизации робота относительно текущего целевого состояния, и логического сетевого оператора, описывающего систему логического вывода, которая по состоянию летающего робота относительно пространственной целевой точки траектории осуществляет или не осуществляет переключение на новую целевую точку.

По сравнению с методом вариационного генетического программирования [32] метод вариационного аналитического программирования [19, 23] имеет более компактный код записи математического выражения и меньшее количество малых вариаций базисного решения и более компактную форму описания малых вариаций. В частности, метод вариационного генетического программирования требует для кодирования математического выражения упорядоченное множество целочисленных двух компонентных векторов. Метод вариационного

аналитического программирования требует упорядоченное множество целых чисел.

Цель и задачи диссертационной работы.

Цель диссертационного исследования заключается в разработке методики решения задачи мониторинга местности группой летающих роботов и в разработке и исследовании нового вычислительного метода вариационного аналитического программирования для решения задачи синтеза системы управления.

Для достижения поставленной цели в диссертационной работе решались следующие **задачи**:

- Решение задачи группы коммивояжеров в трехмерном пространстве для определения оптимальных траекторий для каждого летающего робота.
- Разработка метода вариационного аналитического программирования для синтеза системы управления;
- применение метода вариационного аналитического программирования для решения задачи синтеза системы управления;
- решение задачи синтеза системы управления летающим роботом методом вариационного аналитического программирования;
- исследование качества синтезированной системы управления летающим роботом с целью определения сохранения свойств системы при изменении начальных условий и наличии внешних возмущений.

Объектом исследования является: система управления мобильным летающим роботом типа квадрокоптер.

Предметом диссертационного исследования является: система управления летающим роботом, определенная в форме математической модели объекта управления, летающего робота и требований постановки задачи синтеза: функционала качества, терминальных и начальных условий, ограничений на управление.

Метод проведения исследования. Для проведения исследований были использованы: методы теории управления, метод функционального анализа, методы многокритериальной оптимизаций, численный метод интегрирования дифференциальных уравнений, численные методы оптимизации, эволюционные вычислительные методы, методы символьной регрессии.

Методология и методы исследования состоят в проведении большого количества вычислительных экспериментов, выполняемых при синтезе систем автоматического управления для каждого возможного решения, генерируемого и модифицируемого эволюционным алгоритмом.

Достоверность полученных в диссертации результатов подтверждается программной реализацией разработанных алгоритмов и вычислительными экспериментами, проводимыми в процессе решения задач синтеза управления методом вариационного аналитического программирования.

Научная новизна диссертационной работы состоит:

- В разработке нового численного метода вариационного аналитического программирования, который отличается от известного метода символьной регрессии аналитического программирования, использованием принципа малых вариаций базисного решения;
- В разработке генетического алгоритма для решения задачи синтеза системы управления методом вариационного аналитического программирования;
- В решении задачи синтеза системы управления летающим роботом методом вариационного аналитического программирования;
- В решении задачи группы коммивояжеров вариационным генетическим алгоритмом.

Практическая значимость диссертационной работы состоит в разработке алгоритмов и комплекса программ для автоматического синтеза систем управления методом вариационного аналитического программирования. Работоспособность программного комплекса подтверждена вычислительными

экспериментами. Полученные в результате автоматического синтеза математические выражения для системы управления и могут быть непосредственно использованы на бортовых компьютерах систем управления летающими роботами.

Теоретическая значимость диссертационной работы заключается в исследовании метода вариационного аналитического программирования, в частности, в исследовании и построении малых вариаций структуры данных, кодирующих искомое математическое выражение, а также в сравнительном анализе различных методов символьной регрессии, применяемых для решения задачи синтеза управления, метода сетевого оператора, метода аналитического программирования и метода вариационного аналитического программирования.

Внедрение результатов диссертационной работы.

Разработанные методы и алгоритмы использовались в учебном процессе Инженерной академии Российского Университета Дружбы Народов в департаменте «Механики и мехатроники» при подготовке магистрантов по направлению «Управление в технических системах», при чтении курсов «Современные инструментальные средства интеллектуальных систем», «Современные проблемы теории управления», «Автоматизированное проектирование средств и систем управления», а также в учебном процессе кафедры «Системный анализ и управление» Аэрокосмического института МАИ при чтении курсов «Динамическое проектирование системы управления летательного аппарата», «Оптимальное управление летательными аппаратами».

Внедрение результатов диссертационных исследований подтверждены соответствующими актами.

Апробация работ.

Результаты диссертационных исследований апробированы на применении их в задаче синтеза системы управления реального объекта, летающего робота. Полученные в диссертации результаты докладывались на следующих конференциях:

- VII Международная научно-практическая конференция «Инженерные системы - 2014». Москва, РУДН, 16 - 18 апреля 2014 г.
- международный симпозиум «Надежность и качество», Пенза - 26 мая – 01 июня 2014 г.
- одиннадцатый международный симпозиум «Интеллектуальные системы INTELS'2014», Москва, 30 июня – 04 июля, 2014 г.
- школа-семинар молодых ученых «Фундаментальные проблемы системной безопасности», Елец, 20-22 ноября 2014 г.
- XII International Symposium «Intelligent Systems-2016», INTELS'2016, October 5-7, 2016 Moscow, Russia.
- 2017 11th IEEE International Conference on Application of Information and Communication Technologies (AICT) 20-22 September 2017, Moscow, Russia.
- XXIII Международной научной конференции «Системный анализ, управление и навигация» (г. Евпатория, 2018 г.)

Все исследования, результаты которых приведены в диссертации, **получены лично автором**. При использовании чужих научных результатов в диссертации привендены ссылки на использованные источники.

Публикации.

Основные результаты диссертации опубликованы в 22 печатных работах, в том числе в 4 статьях в рецензируемых изданиях, рекомендованных ВАК, и в 4 статьях в трудах международных конференций, включенных в базы цитирования SCOPUS и Web of Science. В совместных работах результаты принадлежат соавторам в равных долях.

Основные научные положения, выносимые на защиту.

На защиту выносятся следующие научные результаты, полученные в процессе выполнения диссертационных исследований:

- Алгоритм синтеза системы автоматического управления методом вариационного аналитического программирования.

- Применение метода вариационного аналитического программирования для решения задачи синтеза системы управления.
- Решение задачи синтеза системы управления летающим роботом методом вариационного аналитического программирования.
- Решение задачи группы коммивояжеров вариационным генетическим алгоритмом.

Структура и объём работы.

Диссертация состоит из введения, трех глав, основных выводов по работе, списка цитируемой литературы, содержащего 87 наименований. Работа изложена на 109 листах, включая 31 рисунок.

Во **введении** обоснована актуальность темы, сформулированы предмет, цель и задачи исследования, методы исследования, новизна научных результатов и практическая значимость полученных результатов, основные положения, выносимые на защиту, приведены данные о структуре и объеме диссертации.

В первой главе диссертации представлена постановка задачи синтеза управления группой летающих роботов и ее формальное описание, приведен обзор известных аналитических и численных методов ее решения, приведено описание численного метода сетевого оператора для решения задачи синтеза управления и метода аналитического программирования, приведено описание решения задачи для группы коммивояжеров вариационным генетическим алгоритмом.

Во второй главе диссертации приведено описание метода вариационного аналитического программирования, приведено описание структуры данных для кодирования математического выражения, приведено описание малых вариаций структуры данных, приведено описание генетического алгоритма поиска решения с использованием принципа малых вариаций базисного решения, приведен пример решения задачи синтеза системы управления для объекта управления третьего порядка с фазовыми ограничениями методом вариационного аналитического программирования.

В третьей главе диссертации приведено описание решения задачи мониторинга местности для группы летающих роботов. В главе приведено решение задачи группы коммивояжеров для 23 точек трехмерного пространства и двух квадрокоптеров. На первом этапе решена задача поиска оптимальных траекторий для каждого квадрокоптера с помощью вариационного генетического алгоритма. Сформулирована и решена с помощью разработанного в диссертации метода вариационного аналитического программирования задача синтеза системы управления квадрокоптером. Решение задачи обеспечивает устойчивость квадрокоптера относительно точки в трехмерном пространстве. Приведены результаты вычислительного эксперимента для управления движением двух летающих роботов по двум пространственным траекториям.

В заключении диссертации сформулированы основные научные результаты, полученные в диссертационной работе.

1. ЗАДАЧА СИНТЕЗА УПРАВЛЕНИЯ ГРУППОЙ КВАДРОКОПТЕРОВ И ЕЕ РЕШЕНИЕ МЕТОДОМ ВАРИАЦИОННОГО АНАЛИТИЧЕСКОГО ПРОГРАММИРОВАНИЯ.

Задача синтеза системы управления заключается в нахождении функциональной зависимости управления от состояния объекта управления. В задаче синтеза управления задают модель объекта управления в виде системы обыкновенных дифференциальных уравнений, область начальных значений для данной системы, ограничения на управление, целевое или терминальное многообразие и критерий качества управления в форме функционала. Искомая функция должна обеспечивать возможность определения значений вектора управления по значениям вектора состояния объекта управления. Данную функцию называют синтезирующей функцией или функцией управления. Вычисленное по синтезирующей функции управление обеспечивает для решения системы дифференциальных уравнений модели объекта управления попадание из любого начального состояния заданной области на терминальное многообразие с оптимальным значением критерия качества управления.

Формулировка задачи синтеза управления появилась в пятидесятые годы. Данная задача стала естественным продолжением исследований по развитию систем регулирования. Подавляющее большинство систем регулирования использует принцип обратной связи. Регулятор компенсирует величину отклонения выходного сигнала от желаемого значения. Полученная в результате решения задачи синтеза система управления использует информацию о состоянии объекта, терминальном положении и оценку критерия качества для вычисления оптимального значения управляющего воздействия.

Наиболее известным результатом в пятидесятые годы в области синтеза управления был вывод уравнения Беллмана [1, 3, 11, 31, 32, 33, 41, 49, 50, 58]. Данное уравнение выводилось с допущениями неограниченного управления и дифференцируемости правых частей математической модели объекта управления по вектору состояния. Уравнение Беллмана включает функцию минимума и

представляет собой дифференциальное уравнение в частных производных. Решением уравнения Беллмана является многомерная гладкая функция координат вектора пространства состояний.

Вычисление минимума в уравнении Беллмана возможно только при наличии одного начального состояния, поэтому уравнение не предусматривает решение задачи синтеза для области начальных условий. Уравнение Беллмана в общем случае не имеет аналитического решения. Полученное в результате решения уравнения Беллмана гладкое управление не всегда является оптимальным. В некоторых практических задачах, например в задаче быстрогодействия, оптимальное управление должно иметь скачкообразное изменение, т.е. функция управления должна содержать разрывы первого рода.

Следующим важным результатом в области синтеза управления было решение уравнения Беллмана для объекта управления, описываемого системой линейных автономных уравнений, и квадратичного критерия качества. Данное решение приводит к получению линейного регулятора и называется методом аналитического конструирования оптимальных регуляторов (АКОР) [8, 9, 32, 45, 48, 58]. Метод АКОР для решения задачи стабилизации позволяет получить оптимальный регулятор для любого начального условия в окрестности стабилизируемого состояния.

В начале шестидесятых годов был сформулирован принцип максимума Л.С. Понтрягина [1, 2, 11, 58–60]. Данный результат определяет необходимые условия для решения задачи оптимального управления. Задача оптимального управления отличается от задачи синтеза управления. Задача оптимального управления требует нахождения управления в виде функции времени. Задача синтеза управления требует нахождения управления в виде функции от координат вектора пространства состояний объекта управления. Применение принципа максимума иногда позволяет решить задачу синтеза управления. Известны решения задачи синтеза управления по критерию быстрогодействия [9]. Принцип максимума в этих задачах позволил найти области в пространстве состояний, где управление принимает предельные значения. Синтезирующая функция в этих

задачах определяет границы областей или условия переключения значений управления. Решение этих задач синтеза приводит к нахождению разрывного управления и справедливо на всем пространстве состояний.

К аналитическим методам решения задачи синтеза управления следует отнести применение функции А.М. Ляпунова [33, 46, 47] для задач обеспечения устойчивости объекта управления. В процессе исследований задач управления была доказана связь между функциями Беллмана и функцией Ляпунова. Во многих работах по синтезу управления решение уравнения называют функцией Беллмана – Ляпунова.

К последним аналитическим результатам в области синтеза следует отнести синергетическую теорию. Результатом применения данной теории к решению задачи синтеза явилось создание Таганрогским профессором А.А. Колесниковым метода аналитического конструирования агрегированных регуляторов (АКАР) [39, 40]. Суть данного метода заключается в умозрительном или интуитивном построении притягивающих многообразий в пространстве состояний объекта управления. Для построения многообразий исследователь должен знать или предполагать приблизительную форму оптимальных траекторий объекта в пространстве состояний. По форме оптимальных траекторий исследователь строит многообразие в пространстве состояний. Каждому уравнению многообразия ставится в соответствие некоторая агрегированная переменная. Далее строится для каждой агрегированной переменной устойчивое дифференциальное уравнение. Обычно строят линейное дифференциальное уравнение первого порядка. Решение этого уравнения из любого значения агрегированной переменной стремится к нулю. Нулевые значения всех агрегированных переменных соответствуют нахождению объекта управления на заданном оптимальном многообразии. Производная агрегированной переменной по времени включает производную по вектору состояния и правые части дифференциальных уравнения модели объекта управления. В правые части модели входит вектор управления. Разрешение дифференциальных уравнений для агрегированных переменных относительно вектора управления приводит к

построению синтезирующей функции управления. Метод АКАР сложно реализовать на практике. Для реализации данного метода необходимо знать оптимальное поведение объекта управления. При разрешении уравнений агрегированных переменных относительно управления не учитываются ограничения на управление.

Все аналитические методы синтеза управления в основном направлены на решение задачи обеспечения устойчивости или стабилизации, не учитывают разрывность синтезирующей функции и могут применяться только для не сложных объектов малой размерности.

В практических задачах исследователи используют свойства объекта управления, функционала, ограничений и на основании исследуемых специальных характеристик строят систему управления на по принципу обратной связи. Такой подход сегодня наиболее популярен. В общем случае назовем его методом технического синтеза [34–37, 52, 54, 55, 66].

В последнее время с появлением эволюционных методов вычисления появилась возможность численного решения задачи синтеза управления. Исследованию одного из таких численных методов, метода вариационного аналитического программирования, на примере решения прикладных задач синтеза управления посвящена настоящая работа. Обзор существующих численных методов приведем позже после формальной постановки задачи численного синтеза управления.

1.1. Формальная постановка задачи

Рассмотрим задачу мониторинга местности группой квадрокоптеров. Для проведения мониторинга в трехмерном пространстве задано задействовано некоторое количество квадрокоптеров, которые должны пройти заданное количество точек. Считаем, что точки, через которые должны пройти квадрокоптеры, определены, но какой из квадрокоптеров должен пройти через какие точки первоначально не известно. В каждой точке должен побывать хотя бы один из квадрокоптеров. Каждая точка определена тремя координатами.

Предполагаем, что все квадрокоптеры стартуют из заданного начального состояния и после прохождения выбранных для него точек возвращается обратно в то же место, откуда он стартовал.

На первом этапе решаем задачу отбора точек для каждого квадрокоптера. Считаем, что все точки пронумерованы и каждый из квадрокоптеров должен пройти приблизительно одинаковое количество точек. Первый квадрокоптер проходит первые K точек, следующий другие K точек и т.д. Необходимо так перенумеровать точки, чтобы суммарный путь всех квадрокоптеров был минимальным. Эта задача является разновидностью задачи коммивояжера и относится к трудным вычислительным задачам, для которых не известны полиномиальные алгоритмы решения. Для решения этой задачи используем эволюционный генетический алгоритм, которые не находят гарантированно оптимального решения, но позволяет за конечное количество итераций найти близкое к оптимальному решение.

На втором этапе решаем задачу синтеза управления квадрокоптером. В задаче синтеза управления находим нелинейную обратную связь в виде многомерной функции, аргументами которой являются компоненты вектора отклонения состояния квадрокоптера от заданной точки в пространстве состояний. Решение данной задачи обеспечивает стабилизацию квадрокоптера относительно точки в пространстве состояний. При решении задачи синтеза используем новый метод символьной регрессии, метод вариационного аналитического программирования, который является основным теоретическим результатом настоящей работы.

Сформулируем постановки решаемых задач, задачи поиска траекторий движения квадрокоптеров, и задачу синтеза управления.

1.2. Задача поиска оптимальных пространственных траекторий

Пусть задано первоначальные положения квадрокоптеров.

$$P_0 = \{p^{0,1}, \dots, p^{0,N}\}, \quad (1.1)$$

где $\mathbf{p}^{0,i}$ - вектор координат точки трехмерного пространства, $\mathbf{p}^{0,i} = [x^{0,i} \ y^{0,i} \ z^{0,i}]^T$, $i = 1, \dots, N$, N - количество квадрокоптеров.

Задано множество координат точек мониторинга в трехмерном пространстве

$$P = \{\mathbf{p}^1, \dots, \mathbf{p}^M\}, \quad (1.2)$$

где $\mathbf{p}^i = [x_i \ y_i \ z_i]^T$, $i = 1, \dots, M$, $M > N$.

Необходимо найти N подмножеств точек из (1.2)

$$P_1 \cup \dots \cup P_N = P, \quad (1.3)$$

где

$$P_1 = (\mathbf{p}^{\alpha_{1,1}}, \dots, \mathbf{p}^{\alpha_{1,K_1}}), \dots, P_N = (\mathbf{p}^{\alpha_{N,1}}, \dots, \mathbf{p}^{\alpha_{N,K_N}}), \quad (1.4)$$

$$\sum_{i=1}^N K_i = M. \quad (1.5)$$

Разбиение множеств (1.2) выполняем по условию минимума функционала

$$\sum_{i=1}^N |P_i| \rightarrow \min, \quad (1.6)$$

где

$$|P_i| = \sqrt{x_{\alpha_i,1}^2 + y_{\alpha_i,1}^2 + z_{\alpha_i,1}^2} + \sum_{j=1}^{K_i-1} \|\mathbf{p}^{\alpha_i,j+1} - \mathbf{p}^{\alpha_i,j}\|_2 + \sqrt{x_{\alpha_i,K_i}^2 + y_{\alpha_i,K_i}^2 + z_{\alpha_i,K_i}^2} \rightarrow \min, \quad (1.7)$$

$$\|\mathbf{p}^A - \mathbf{p}^B\|_2 = \sqrt{(x_A - x_B)^2 + (y_A - y_B)^2 + (z_A - z_B)^2}. \quad (1.8)$$

1.3. Задача синтеза управления

Задана математическая модель объекта управления [23]

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}), \quad (1.9)$$

где \mathbf{x} - вектор состояния объекта управления, $\mathbf{x} = [x_1 \dots x_n]^T$, $\mathbf{x} \in \mathbb{R}^n$, \mathbf{u} - вектор управления, $\mathbf{u} = [u_1 \dots u_m]^T$, $\mathbf{u} \in U \subseteq \mathbb{R}^m$, $m \leq n$, U - ограниченное замкнутое множество.

Задано множество начальных значений

$$X_0 \subseteq \mathbb{R}^n. \quad (1.10)$$

Заданы терминальные условия или цель управления

$$\varphi_i(\mathbf{x}) = 0, \quad i = \overline{1, r}. \quad (1.11)$$

Задан критерий качества управления в виде функционала

$$J = \int_0^{t_f} f_0(\mathbf{x}(t), \mathbf{u}(t)) dt \rightarrow \min, \quad (1.12)$$

где t_f - время управления, которое может быть не задано и определяться по условию (1.3) достижения цели управления.

Необходимо найти управление в виде многомерной функции от компонент вектора пространства состояний

$$\mathbf{u} = \mathbf{h}(\mathbf{x}). \quad (1.13)$$

Функция $\mathbf{h}(\mathbf{x})$ должна обладать следующими свойствами. Функция $\mathbf{h}(\mathbf{x})$ должна удовлетворять ограничениям на управление

$$\mathbf{h}(\mathbf{x}) \subseteq U, \quad \forall \mathbf{x} \in \mathbb{R}^n. \quad (1.14)$$

При подстановке функции $\mathbf{h}(\mathbf{x})$ в модель объекта управления (1.9) получаем систему дифференциальных уравнений

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

Полученная система уравнений для любого начального значения из заданной области (1.10) начальных значений

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

имеет решение в форме векторной функции времени $\mathbf{x}(t, \mathbf{x}^0)$. Данное решение за конечное время $t_f < \infty$ достигает цели управления (1.11)

$$\varphi_i(\mathbf{x}(t_f, \mathbf{x}^0)) = 0, \quad i = \overline{1, r}, \quad \mathbf{x}^0 \in X_0, \quad (1.15)$$

и обеспечивает минимум функционалу (1.12) качества управления на множестве всех допустимых управлений $\tilde{U} = \{\tilde{\mathbf{u}}(\cdot)\}$ с учетом ограничений $\tilde{\mathbf{u}}(t) \subseteq U, 0 \leq t \leq t_f$,

$$\int_0^{t_f} f_0(\mathbf{x}(t), h(\mathbf{x}(t))) dt = \min_{\mathbf{u} \in \tilde{U}} \int_0^{t_f} f_0(\mathbf{x}(t), \mathbf{u}(t)) dt. \quad (1.16)$$

Решение задачи (1.9) - (1.16) синтеза управления требует нахождения многомерной синтезирующей функции управления (1.13), вычисление значения функционала (1.12) и проверку выполнения терминальных условий (1.11). Для любого начального состояния из заданной области (1.10) величина функционала (1.12) должна быть наименьшей и терминальные условия (1.11) должны выполняться.

Пусть для одного начального условия $\mathbf{x}^{0,i} \in X_0$ построено возможное решение задачи синтеза $\mathbf{h}(\mathbf{x})$. Возникает проблема проверки этого решения. Решим для этого же начального условия $\mathbf{x}^{0,i}$ задачу оптимального управления и найдем управление в виде функции времени $\mathbf{u}(t)$. Для оценки оптимальности возможного решения $\mathbf{h}(\mathbf{x})$ используем следующие соотношения:

$$\|\mathbf{u}(\cdot) - \mathbf{h}(\mathbf{x}(\cdot))\| = \Delta_u, \quad (1.17)$$

$$\left| \int_0^{t_f} f_0(\mathbf{x}(t), h(\mathbf{x}(t))) dt - \int_0^{t_f} f_0(\mathbf{x}(t), \mathbf{u}(t)) dt \right| = \Delta_J, \quad (1.18)$$

$$\sum_{k=1}^r \left| \varphi_k(\mathbf{x}(t_f, \mathbf{x}^{0,i})) \right| = \Delta_f, \quad (1.19)$$

где $\mathbf{u}(\cdot)$ - решение задачи оптимального управления, $\mathbf{h}(\mathbf{x}(\cdot))$ - полученное по синтезирующей функции управление, $\mathbf{u}(t)$ - значение оптимального управления в момент времени t , $h(\mathbf{x}(t))$ - значение синтезированного управления в момент времени t .

Близость к нулю всех показателей (1.17) – (1.19) для одного начального состояния $\mathbf{x}^{0,i} \in X_0$ не гарантирует выполнение этих условий для другого начального состояния $\mathbf{x}^{0,j} \neq \mathbf{x}^{0,i}$ для той же синтезирующей функции $\mathbf{h}(\mathbf{x})$.

Для корректной оценки оптимальности возможного решения $\mathbf{h}(\mathbf{x})$ необходимо вычислять показатели (1.17) для всех других начальных условий из заданной области (1.10). Это требует решения сложных задач оптимального управления для этих же начальных условий. Непрерывность области начальных условий $X_0 \subseteq \mathbb{R}^n$ указывает на содержание в ней бесконечного количества точек и не позволяет использовать корректно оценки (1.17) – (1.19).

Переформулируем задачу (1.9) – (1.16) для численного синтеза управления. Заменяем область начальных условий (1.10) множеством точек

$$\tilde{X}_0 = (\mathbf{x}^{0,1}, \dots, \mathbf{x}^{0,K}), \quad (1.20)$$

где $\mathbf{x}^{0,i} = [x_1^{0,i} \dots x_n^{0,i}]^T$, $i = \overline{1, K}$.

Заменяем функционал (1.4) суммой вычисленных для каждого начального условия функционалов.

$$J_1 = \sum_{i=1}^K \left(\int_0^{t_f} f_0(\mathbf{x}(t), \mathbf{u}(t)) dt \right)_{\mathbf{x}^{0,i}} \rightarrow \min, \quad (1.21)$$

где $(J)_{\mathbf{x}^{0,i}}$ указывает на вычисление функционала J при начальных значениях $\mathbf{x}(0) = \mathbf{x}^{0,i}$.

Требование выполнения терминальных условий (1.11) определим в виде дополнительного функционала

$$J_2 = \max_i \left\| \left(\varphi(\mathbf{x}(t_f)) \right)_{\mathbf{x}^{0,i}} \right\| \rightarrow \min, \quad (1.22)$$

где $\varphi(\mathbf{x}) = [\varphi_1(\mathbf{x}) \dots \varphi_r(\mathbf{x})]^T$,

$$\|\varphi(\mathbf{x})\| = \sqrt{\sum_{j=1}^r \varphi_j^2(\mathbf{x})}.$$

Для задач с незадаанным временем t_f окончания процесса управления терминальный функционал (1.14) имеет вид

$$J_2 = \max_i \min_{t_f} \left\| \left(\varphi(\mathbf{x}(t_f)) \right)_{\mathbf{x}^{0,i}} \right\| \rightarrow \min, \quad (1.23)$$

где $t_f \in [0, t^+]$, t^+ - максимальное допустимое время управления.

В результате получаем задачу многокритериальной оптимизации. Наличие нескольких функционалов требует нахождения множества Парето. Решением задачи синтеза управления является функция $\mathbf{u} = \mathbf{h}(\mathbf{x})$. Множество Парето эффективных решения представляет собой множество функций управления

$$\tilde{\mathbf{P}} = \{ \tilde{\mathbf{h}}^i(\mathbf{x}) : i = 1, 2, \dots \}, \quad (1.24)$$

Множество Парето включает наилучшие решения по значениям функционалов (1.21), (1.22) или (1.23). Для любого не входящего во множество Парето возможного решения всегда найдется решение из множества Парето, которое дает не хуже значение одного функционала и обязательно лучше значение другого функционала: $\forall \mathbf{h}(\mathbf{x}) \notin \tilde{\mathbf{P}}, \exists \tilde{\mathbf{h}}^i(\mathbf{x}) \in \tilde{\mathbf{P}}$, что

$$(J_1(\tilde{\mathbf{h}}^i(\mathbf{x})) < J_1(\mathbf{h}(\mathbf{x}))) \wedge (J_2(\tilde{\mathbf{h}}^i(\mathbf{x})) \leq J_2(\mathbf{h}(\mathbf{x})))$$

или

$$(J_1(\tilde{\mathbf{h}}^i(\mathbf{x})) \leq J_1(\mathbf{h}(\mathbf{x}))) \wedge (J_2(\tilde{\mathbf{h}}^i(\mathbf{x})) < J_2(\mathbf{h}(\mathbf{x}))),$$

где $J_\alpha(A)$ - значение функционала J_α , $\alpha = 1, 2$, для решения A , $A \in \{\mathbf{h}(\mathbf{x}), \tilde{\mathbf{h}}^i(\mathbf{x})\}$.

1.4. Математическая модель квадрокоптера

Квадрокоптер осуществляет полет и перемещение в пространстве за счет сил тяги винтов. Изменение соотношений тяг каждого из винтов позволяет изменить углы направления общей силы тяги, проекции которой на оси инерциальной системы координат осуществляют перемещения у центра масс квадрокоптера в пространстве.

На рис.1.1 приведена упрощенная модель квадрокоптера.

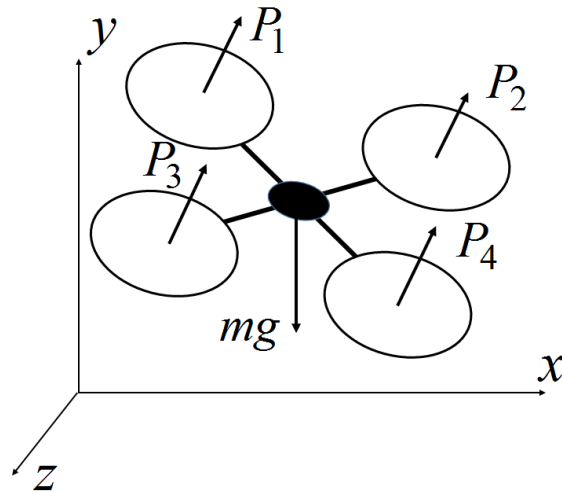


Рис. 1.1. Упрощённая модель квадрокоптера

Сила тяги в нормальной земной системе координат:

$$P = [P_x, P_y, P_z]^T = R(\gamma, \vartheta, \psi)P_w + P_g + P_f, \quad (1.25)$$

где $R(\gamma, \vartheta, \psi)$ – матрица перехода; ψ, ϑ, γ – углы рыскания, тангажа, крена, P_w – сила тяги четырех винтов квадрокоптера, P_g – сила тяжести, P_f – сила сопротивления воздуха,

$$P_g = [0 \sum_{i=1}^4 P_i \ 0]^T, \quad (1.26)$$

P_i – сила тяги винта i , $i = 1, 2, 3, 4$.

Проекции силы тяжести в нормальной системе координат имеют следующие значения

$$P_g = [0 \ -mg \ 0]^T. \quad (1.27)$$

Силой сопротивления воздуха в дальнейших расчетах принимаем равной нулю,

$$P_f = [0 \ 0 \ 0]^T. \quad (1.28)$$

Матрица перехода имеет следующий вид:

$$R(\gamma, \vartheta, \psi) = \begin{bmatrix} \cos \vartheta \cos \psi & \cos \psi \sin \vartheta \sin \gamma - \sin \psi \sin \gamma & \cos \psi \sin \vartheta \cos \gamma + \sin \psi \sin \gamma \\ \sin \psi \cos \vartheta & \sin \psi \sin \vartheta \sin \gamma + \cos \psi \cos \gamma & \sin \psi \sin \vartheta \cos \gamma - \cos \psi \sin \gamma \\ -\sin \vartheta & \cos \vartheta \sin \gamma & \cos \vartheta \cos \gamma \end{bmatrix}. \quad (1.29)$$

Уравнение динамики движения центра масс в нормальной земной системе координат:

$$\ddot{x} = P_x / m; \quad \ddot{y} = P_y / m; \quad \ddot{z} = P_z / m, \quad (1.30)$$

С учётом симметрии аппарата и считая, что центр масс расположен в начале координат связанной системы, уравнения динамики углового движения в связанной системе координат записываем в виде:

$$\begin{cases} \dot{w}_x = w_y w_z (I_y - I_z) / I_x + M_{Rx} / I_x \\ \dot{w}_y = w_x w_z (I_z - I_x) / I_y + M_{Ry} / I_y \\ \dot{w}_z = w_x w_y (I_x - I_y) / I_z + M_{Rz} / I_z \end{cases} \quad (1.31)$$

$$\begin{cases} M_{Rx} = M_{qx} + M_{mx} + M_{px} + M_{wx} + M_{gx} \\ M_{Ry} = M_{qy} + M_{my} + M_{py} + M_{wy} + M_{gy} \\ M_{Rz} = M_{qz} + M_{wz} + M_{gz} \end{cases}, \quad (1.32)$$

где w_x , w_y , w_z – проекции вектора угловой скорости аппарата на связанную систему координат; M_{Rx} , M_{Ry} , M_{Rz} – проекции результирующего момента; I_x , I_y , I_z – осевые моменты инерции аппарата; M_{qx} , M_{qy} , M_{qz} – моменты, создаваемые винтами, M_{mx} , M_{my} и M_{px} , M_{py} – гироскопические моменты двигателей и винтов, M_{wx} , M_{wy} , M_{wz} – моменты, создаваемые из-за воздействия ветра, M_{gx} , M_{gy} , M_{gz} – моменты, создаваемые силами тяги эффекта Земли. Если пренебречь инерционностью винтов при изменении угловых скоростей их вращения, то указанные моменты можно выразить следующим образом:

$$\begin{cases} M_{px} + M_{wx} + M_{gx} = (P_4 - P_2) \cdot l \\ M_{qy} + M_{wy} + M_{gy} = (P_3 - P_1) \cdot l \\ M_{pz} + M_{wz} + M_{gz} = M_1 + M_2 + M_3 + M_4 \\ M_i = m_{Pi} w_i^2 \end{cases}, \quad (1.33)$$

$$\begin{cases} M_{mx} = I_m w_y (w_2 + w_4 - w_1 - w_3) \\ M_{my} = I_m w_x (w_1 + w_3 - w_2 - w_4) \end{cases},$$

$$\begin{cases} M_{px} = I_p w_y (w_2 + w_4 - w_1 - w_3) \\ M_{py} = I_p w_x (w_1 + w_3 - w_2 - w_4) \end{cases},$$

где l – расстояние от центра масс до оси винта, I_m и I_p – моменты инерции ротора и винта; m_{Pi} – коэффициент момента.

Изменения углов Эйлера определяются через проекции угловой скорости кинематическими уравнениями Эйлера:

$$\begin{cases} \dot{\gamma} = \omega_x + (\omega_y \sin \gamma + \omega_z \cos \gamma) \tan \vartheta \\ \dot{\theta} = \omega_y \cos \gamma + \omega_z \sin \gamma \\ \dot{\psi} = (\omega_y \sin \gamma + \omega_z \cos \gamma) / \cos \vartheta \end{cases} \quad (1.34)$$

Задачу траекторного управления полётом квадрокоптера можно рассмотреть как последовательность задач перелёта в очередную заданную точку маршрута или движения по заданным участкам типовых траекторий (например, по прямой или по дуге окружности). В этом случае система автоматического управления может быть построена как система с обратной связью, осуществляющая отслеживание заданного маршрута. При этом можно выделить канал управления высотой и канал управления движением в горизонтальной плоскости. Стабилизация и управление в вертикальном направлении обеспечивается изменением суммарной величины тяги. Горизонтальное перемещение аппарата происходит под действием горизонтальной проекции суммарного вектора тяги, отклонённого от вертикали. В рассматриваемом варианте отклонение вектора тяги происходит за счёт изменения углов тангажа и крена при фиксированном положении угла рыскания. Изменение углового положения достигается путём дифференцированного управления скоростями вращения винтов, дающего соответствующие различия их сил тяги и моментов. Подсистему, обеспечивающую необходимые значения угловых параметров и высоты за счёт изменения тяги винтов, можно назвать системой ориентации и стабилизации, а подсистему, осуществляющую отслеживание заданных траекторий, – системой траекторного управления. Нужно отметить, что рассматриваемый вариант не является самым эффективным, но в нём наиболее наглядно реализуется разделение управления по каналам.

1.5. Методы решения задачи поиска оптимальных траекторий

Задача для группы коммивояжеров существенно не отличается от классической задачи коммивояжера. В нашем случае к целевой функции, которая оценивает длину пути квадрокоптеров, добавляется путь от начальной точки до первого пункта пути и обратно из последнего пункта в точку, куда должен попасть квадрокоптер. По методам решения задач коммивояжера написано достаточно много работ. Задача относится к классу NP-трудных [44] задач, для которых сегодня не известны алгоритмы с полиномиальной скоростью сходимости. Обзор по классическим методам решения задачи коммивояжера приведен в работе [51]. Попытки эффективно решить данную задачу классическими методами, методом динамического программирования, методом ветвей и границ, методом отсекающих плоскостей и с помощью введения множителей Лагранжа [51] показали, что для больших задач коммивояжера, в которых число пунктов около сотни, нахождение оптимального решения требует существенных затрат времени. В последнее время NP-полные задачи решают с помощью эволюционных алгоритмов, которые не гарантируют нахождение оптимального решения за ограниченное количество итераций, но как правило находят решения, близкое к оптимальному. Известны решения задачи коммивояжера наиболее известным из эволюционных алгоритмов генетическим алгоритмом [79]. Заметим, что при использовании операции скрещивания генетического алгоритма непосредственно к различным путям движения, состоящим из списка пунктов, приводит к повторению пунктов в пути, поэтому в работе [79] было предложена специальная операция скрещивания для задачи коммивояжера. Наиболее часто при решении задачи коммивояжера эволюционным алгоритмом применяют алгоритм муравьиной колонии [74], который, скорее всего, и был специально разработан для подобных задач, в которых требуется нахождение оптимальных путей на графе. В алгоритме случайно прокладывается большое количество путей, и те пути, которые имеют хорошие значения целевой функции, более часто используются специально отмечаются некоторой числовой оценкой, называемой в алгоритме феромоном.

При каждом прохождении пути феромоны увеличиваются на константу, а при отсутствии пути уменьшаются. В результате после определенного количества итераций отбирается наилучший путь, который имеет наилучшую оценку, и элементы которого наиболее часто использовались в процессе поиска.

В диссертации для решения задачи группы коммивояжеров мы используем вариационный генетический алгоритм, который за счет правильного выбора малых вариаций обеспечивает отсутствие повторов пунктов при выполнении операции скрещивания. Успех работы вариационного генетического алгоритма зависит от удачного выбора базисного решения. В работе при выборе базисного решения используем жадный алгоритм, который находит путь с помощью выбора следующего пункта по критерию близости к текущему пункту.

1.6. Обзор численных методов решения задачи синтеза управления

Для численного решения задачи синтеза управления необходимо с помощью вычислительного алгоритма строить структуру и находить параметры многомерной функции. До последнего времени в задачах поиска функции исследователь задавал структуру функции с точностью до значений параметров. Вычислительный алгоритм находил оптимальные значения параметров. Такие методы в основном использовали для аппроксимации экспериментальных данных с помощью различных рядов. Для задачи синтеза необходимо строить многомерную функцию. Использование многомерных рядов приводит к поиску большого количества параметров.

Для аппроксимации многомерных функций используют созданные в шестидесятые годы нейронные сети. Основная проблема использования нейронных сетей заключается в ее обучении. Под обучением нейронной сети понимают нахождение оптимальных значений параметров. В настоящее время появились эффективные алгоритмы обучения нейронной сети. Нейронные сети представляют собой многомерные функции с заданной структурой и неизвестными значениями параметров. Структура нейронной сети позволяет без

особой сложности увеличивать ее размерность и количество настраиваемых параметров.

Существенным результатом для создания численных методов синтеза и алгоритмов поиска структур функций является появление в девяностые годы генетического программирования. Данный метод был разработан для решения задачи автоматического написания программ. Метод генетического программирования основывается на применении эволюционного генетического алгоритма к используемой в трансляторах универсальной польской записи программы в форме строки символов. Усовершенствование метода генетического программирования при решении различных прикладных задач привело к созданию новых методов: грамматической эволюции, аналитического программирования, сетевого оператора, вариационного генетического программирования и вариационного аналитического программирования. Все эти методы называются методами символьной регрессии. Методы позволяют находить структуру и параметры формул математических выражений и могут использоваться для создания численных методов синтеза управления.

Перечисленные методы символьной регрессии позволяют находить структуры и параметры математических выражений, описывающих функциональные зависимости между количественными оценками исходных данных и результатов анализа. Данные методы успешно применяются для решения задач, в которых необходимо найти математические выражения оптимальные по заданным критериям. Такие задачи возникают в области управления роботизированными техническими объектами для синтеза управления

Нейронные сети [12, 38] представляют собой универсальный метод аппроксимации многомерных функций. Нейронные сети применяются в задачах прогнозирования, аппроксимации, классификации, распознавания образов и т.п. В области задач управления нейронные сети широко применяют в адаптивных системах для настройки параметров системы управления.

Существует большое количество видов нейронных сетей. Обобщенно нейронные сети по структуре можно разбить на три класса: сеть с прямыми связями, сеть с перекрестными связями и сеть с обратными связями.

Каждая ячейка сети содержит сумматор. Входные сигналы складываются с определенными весовыми коэффициентами. Нахождение значений весовых коэффициентов называют обучением нейронной сети. На выходе каждой ячейки сигнал подвергается нелинейному преобразованию. Нелинейная функция называется функцией активации и используется для многослойных сетей. Многослойные сети применяют для представления сложных функций. В многослойных сетях выходы предыдущего слоя являются входами следующего.

В качестве функций активации используют гладкие или элементарные линейные функции.

Математическое выражение для нейронной сети является функцией с заданной структурой и большим количеством настраиваемых параметров. Современные эффективные методы оптимизации, алгоритм обратного распространения ошибки, дельта-правило Видроу - Хоффа, и новые эволюционные, популяционные алгоритмы позволяют эффективно обучать нейронные сети с большим количеством настраиваемых параметров. Потенциально метод нейронных сетей может использоваться для численного решения задачи синтеза управления (1.9), (1.13), (1.14), (1.20) - (1.23).

Метод генетического программирования был создан в 1992 году в университете Стэнфорда профессором Дж. Козой [76–78]. Метод предназначался для решения задачи автоматического написания программ. Метод позволяет искать математические выражения в форме записей кодов операторов языка программирования. Для кодирования математического выражения достаточно использовать коды входящих в математическое выражения функций. Метод генетического программирования для поиска решения в виде закодированной записи математического выражения использует генетический алгоритм.

Используемая в генетическом программировании польская запись [53] представляет собой бесскобочный набор символов, обозначающих операции и

операнды программного кода. В трансляторах используются префиксная и постфиксная записи. Дж. Коза применил основную операцию генетического алгоритма [10, 13, 14], операцию скрещивания, к префиксным строкам символов.

Операция скрещивания в методе генетического программирования существенно отличается от операции скрещивания генетического алгоритма. Точка скрещивания в генетическом программировании в скрещиваемых записях должна указывать на функции с одинаковым числом аргументов. В случае несоблюдения этого условия операция скрещивания не выполняется. После нахождения точек скрещивания для функций с одинаковым количеством аргументов в каждой из записей находят подстроки символов математического подвыражения функций в точке скрещивания. Скрещивание осуществляется обменом подстрок символов. В генетическом программировании коды записей могут иметь различную длину, точки скрещивания могут не совпадать, обмениваемые подстроки также могут различаться по длине.

Метод генетического программирования успешно применяется для решения различных задач поиска структур математических выражений. Метод применялся и для решения задачи синтеза управления [56, 57, 63, 64, 77, 78].

Существует несколько разновидностей метода генетического программирования.

Метод индуктивного генетического программирования использует в качестве базовых функций полиномы. Метод позволяет искать математические выражения гладких функций.

Метод декартового генетического программирования позволяет описывать математические выражения с помощью упорядоченного набора функций. Каждая функция описывается узлом графа сети с одинаковым числом входов. В графе сети имеется заданное число слоев и одинаковое число узлов в каждом слое. Выход узла сети на одном слое не может являться входом узла сети на предыдущем слое. В результате представления математического выражения с помощью такой сети некоторые узлы оказываются не задействованными.

Декартово генетическое программирование позволяет избежать рекурсии при вычислении математического выражения по коду записи.

Метод грамматической эволюции появился в 1998 г. Авторами метода являются ирландские ученые К. Райан и М. Онейл [81, 82]. Метод грамматической эволюции предназначен для поиска программного кода в универсальной форме Бэкуса – Наура. Каждый элемент формы Бэкуса – Наура кодируется восьмибитовым двоичным кодом. Данный код называется кодоном. Операция скрещивания в грамматической эволюции выполняется классическим для генетического алгоритма способом. Находят точки скрещивания в отобранных записях кодов и обменивают оставшиеся после точки скрещивания участки кодов. После расшифровки скрещенных кодонов в методе грамматической эволюции могут возникнуть коллизии. Для разрешения коллизий используют коды терминальных элементов.

Метод грамматической эволюции существенно не отличается от метода генетического программирования. Универсальная форма Бэкуса – Наура позволяет искать математические выражения любой сложности. Искомые математические выражения могут включать арифметические, алгебраические, логические функции. Метод позволяет искать вместе с кодом математического выражения и код программы алгоритма поиска этого выражения. Метод грамматической эволюции использовался для решения задач синтеза управления.

Метод аналитического программирования разработан в 2002 году чешским ученым И. Зелинкой [65, 83–87]. Метод аналитического программирования имеет наиболее компактный код записи математического выражения. Для поиска оптимального решения в методе аналитического программирования используются различные эволюционные алгоритмы.

Настоящая работа посвящена применению метода вариационного аналитического программирования к решению задачи синтеза управления.

Рассмотрим метод аналитического программирования подробнее.

Введем упорядоченные множества функций с определенными количествами аргументов

$$F_i = (f_{i,1}(z_1, \dots, z_i), \dots, f_{i,m_i}(z_1, \dots, z_i)) , i = \overline{0, n}, \quad (1.35)$$

где $f_{i,j}(z_1, \dots, z_i)$ - функция под номером j с количеством аргументов i , $j = \overline{1, m_i}$, $i = \overline{0, n}$.

Объединим все множества (1.35) в одно объединенное множество

$$F = \bigcup_{i=0}^n F_i . \quad (1.36)$$

Пронумеруем элементы объединенного множества

$$F = (f_1, \dots, f_K), \quad (1.37)$$

где

$$K = \sum_{i=0}^n m_i , \quad (1.38)$$

$$f_1 = f_{0,1}, f_2 = f_{0,2}, \dots, f_{m_0} = f_{0,m_0}, f_{m_0+1}(z) = f_{1,1}(z), \dots, f_{m_0+m_1}(z) = f_{1,m_1}(z),$$

$$f_{m_0+m_1+1}(z_1, z_2) = f_{2,1}(z_1, z_2), \dots, f_{m_0+\dots+m_n}(z_1, \dots, z_n) = f_{n,m_n}(z_1, \dots, z_n).$$

Отдельно для разрешения коллизий определим множество функций без аргументов. Данное множество для математических выражений представляет собой множество параметров или переменных

$$F_0 = (f_{0,1}, \dots, f_{0,m_0}) = (x_1, \dots, x_N, q_1, \dots, q_p). \quad (1.39)$$

Запись кода математического выражения метода аналитического программирования представляет собой упорядоченное множество целых чисел

$$C = (c_1, \dots, c_M) , \quad (1.40)$$

где $c_i \in \{1, \dots, K\}$, $i = \overline{1, M}$.

Каждое число c_i указывает номер элемента в объединенном множестве F (1.36). Запись имеет префиксный порядок. Код функции в записи предшествует коду аргумента. Длина записи кода ограничена. Дополнительное множество переменных и параметров используют для корректного завершения записи кода. Рассмотрим пример. Пусть задано множество функций

$$F_0 = (f_{01} = x_1, f_{02} = x_2, f_{0,3} = \alpha, f_{0,4} = \beta) ,$$

$$F_1 = (f_{1,1}(z) = -z, f_{1,2}(z) = e^z, f_{1,3}(z) = |z|, f_{1,4}(z) = \sin(z), f_{1,5}(z) = \cos(z)),$$

$$F_2 = (f_{2,1}(z_1, z_2) = z_1 + z_2, f_{2,2}(z_1, z_2) = z_1 \cdot z_2).$$

Построим объединенное множество

$$F = (f_1 = x_1, f_2 = x_2, f_3 = \alpha, f_4 = \beta, f_5 = -z, f_6 = e^z, f_7 = |z|,$$

$$f_8 = \sin(z), f_9 = \cos(z), f_{10} = z_1 + z_2, f_{11} = z_1 \cdot z_2)$$

Объединенное множество содержит $K = 11$ элементов $|F| = 11$. Множество переменных и параметров содержит четыре элемента $|F_0| = 4$.

Рассмотрим пример кодирования математического выражения. Пусть необходимо закодировать следующее математическое выражение

$$y = -e^{-\alpha x_1} \cos(x_2 + \beta) + \sin|x_2|.$$

Расставим скобки в данном математическом выражении для определения порядка вычислений, получаем

$$y = \left(\left(- \left(e^{-(\alpha x_1)} \times \cos(x_2 + \beta) \right) \right) + \left(\sin(|x_2|) \right) \right).$$

Первой операцией во внешних скобках является операция сложения. Она имеет в объединенном множестве номер 10, $f_{10} = z_1 + z_2$. Первый элемент кода $c_1 = 10$.

Операция сложения имеет два аргумента

$$y = (y_1 + y_2), \quad y_1 = - \left(e^{-(\alpha x_1)} \times \cos(x_2 + \beta) \right), \quad y_2 = \sin(|x_2|).$$

Рассмотрим первый аргумент операции сложения

$$y_1 = - \left(e^{-(\alpha x_1)} \times \cos(x_2 + \beta) \right).$$

Первой операцией данного математического выражения является инверсия знака, функция под номером 5 в объединенном множестве, $c_2 = 5$.

Следующей операцией в математическом выражении для аргумента y_1 является операция умножения, $c_3 = 11$. Далее следует операция экспонента $e^{-(\alpha x_1)}$, $c_4 = 6$. Затем выполняем операцию смены знака $-(\alpha x_1)$, $c_5 = 5$, операцию

умножения $\alpha \times x_1$, $c_6 = 11$. Аргументами последнего умножения являются функции без аргументов, α и x_1 . Данные функции соответствуют элементам f_3 и f_1 в объединенном множестве. Получаем следующие элементы кода: $c_7 = 3$, $c_8 = 1$.

Следующей выполняемой операцией является операция вычисления косинуса, $\cos(x_2 + \beta)$, $c_9 = 9$. Аргументом косинуса является операция сложения функций без аргументов, $x_2 + \beta$, получаем $c_{10} = 10$, $c_{11} = 2$, $c_{12} = 4$.

Вычисляем выражение для аргумента $y_2 = \sin(|x_2|)$. Получаем операцию синуса $c_{13} = 8$. Следующая операция вычисления абсолютной величины $|x_2|$, $c_{14} = 7$. Последний элемент кода x_2 , $c_{15} = 2$.

В результате получаем следующий код записи рассматриваемого примера математического выражения

$$C = (c_1, \dots, c_{15}), C = (10, 5, 11, 6, 5, 11, 3, 1, 9, 10, 2, 4, 8, 7, 2).$$

Для получения математического выражения из записи кода необходимо знать количества элементов в каждом из множеств функций: m_0, \dots, m_n , и количества используемых переменных и параметров N и p . Для рассмотренного примера множества функций имеем максимальное количество аргументов $n = 2$, количество функций без аргументов $m_0 = 4$, количество функций с одним аргументом $m_1 = 5$, количество функций с двумя аргументами $m_2 = 2$, количество переменных во множестве функций без аргументов

$$F_0 = \left(\underbrace{f_{01} = x_1, f_{02} = x_2}_N, \underbrace{f_{0,3} = \alpha, f_{0,4} = \beta}_p \right), N = 2, \text{ количество параметров } p = 2.$$

Количество аргументов функции i и номер функции j определяем по значению элемента c_k кода математического выражения с помощью соотношений

$$i = \begin{cases} 0, & \text{если } 0 \leq c_k \leq m_0 \\ \alpha, & \text{если } \sum_{r=0}^{\alpha-1} m_r \leq c_k \leq \sum_{r=0}^{\alpha} m_r, \alpha = \overline{1, m_n} \end{cases}, \quad (1.41)$$

$$j = c_k - \sum_{r=0}^{i-1} m_r, \quad 1 < i \leq n. \quad (1.42)$$

При соответствии элемента кода c_k функции без аргументов, $i=0$, для определения переменной x_j или параметра q_j необходимо использовать количества используемых переменных N и параметров p .

$$j = \begin{cases} c_k, & \text{если } c_k \leq N \\ c_k - N & \text{иначе} \end{cases}, \quad i=0, \quad (1.43)$$

где код c_k соответствует переменной x_j , если $c_k \leq N$, или параметру q_j , если $N < c_k \leq m_0$.

Для определения корректности записи кода математического выражения используем индекс элемента. Индекс кода элемента указывает на минимальное количество недостающих справа элементов.

Для вычисления индекса элемента c_j используем соотношение

$$T(j) = 1 - j + \sum_{k=1}^j i_k, \quad (1.44)$$

где i_k - количество аргументов функции, соответствующей элементу кода c_k

$$i_k = \begin{cases} 0, & \text{если } 0 \leq c_k \leq m_0 \\ l, & \text{если } \sum_{r=0}^{l-1} m_r \leq c_k \leq \sum_{r=0}^l m_r, l = \overline{1, m_n} \end{cases}. \quad (1.45)$$

Корректная запись кода математического выражения должна удовлетворять соотношениям

$$T(j) > 0, \quad j < M, \quad (1.46)$$

$$T(M) = 0. \quad (1.47)$$

Алгоритм для вычисления математического выражения по записи кода в аналитическом программировании должен располагать информацией о максимальном количестве аргументов в используемых функциях.

В классическом методе аналитического программирования, разработанным И. Зелинкой номер кодируемого элемента вычисляют по позиции в объединенном множестве, но относительно номера предыдущего элемента. Принципиальной разницы между формами кодирования нет. В рассматриваемом примере в коде математического выражения $C = (10, 5, 11, 6, 5, 11, 3, 1, 9, 10, 2, 4, 8, 7, 2)$ первые четыре операции – это операции сложения, инверсии знака, умножения и экспоненты. Согласно кодировке И. Зелинки имеем: код операции сложения $\bar{c}_1 = 10$, код операции инверсии знака вычисляем по положению в объединенном множестве относительно операции сложения, получаем

$$F = \left(\underbrace{x_1, x_2, \alpha, \beta, -z, e^z, |z|, \sin(z), \cos(z)}_5, z_1 + z_2, \underbrace{z_1 \cdot z_2}_1 \right),$$

$$\bar{c}_2 = 1 + 5 = 6.$$

Далее следует операция умножения. Из объединенного множества получаем

$$F = \left(x_1, x_2, \alpha, \beta, -z, \underbrace{e^z, |z|, \sin(z), \cos(z)}_6, z_1 + z_2, z_1 \cdot z_2 \right)$$

$$\bar{c}_3 = 6.$$

Следующая операция вычисления экспоненты имеет код

$$F = \left(\underbrace{x_1, x_2, \alpha, \beta, -z, e^z}_6, |z|, \sin(z), \cos(z), z_1 + z_2, z_1 \cdot z_2 \right)$$

$\bar{c}_4 = 6$. Получаем снова тот же номер, хотя операции отличаются.

Для вычисления относительного кода элемента используем соотношение

$$\bar{c}_1 = c_1, \bar{c}_i = \begin{cases} c_i - c_{i-1}, & \text{если } c_i \geq c_{i-1}, \\ |F| - c_{i-1} + c_i, & \text{иначе} \end{cases}, i = \overline{2, M}. \quad (1.48)$$

Вычисляем относительные позиции остальных элементов рассматриваемого примера математического выражения. В результате получаем код математического выражения в относительных позициях элементов

$$\bar{C} = (10, 6, 6, 6, 10, 6, 3, 9, 8, 1, 3, 2, 4, 10, 6).$$

Сетевой оператор разработан в 2006 году российским профессором А.И. Дивеевым [16–19, 69–71,] специально для решения задачи синтеза системы управления. Метод использует кодировку математических выражений в форме ориентированного графа. В вычислительной машине граф сетевого оператора представляется в виде целочисленной матрицы сетевого оператора. Кодированное математическое выражение состоит из функций без аргументов, с одним аргументом или двумя аргументами.

Для кодирования математического выражения в виде сетевого оператора функции с двумя аргументами должны быть ассоциативны, коммутативны и иметь единичный элемент. Множество функций с одним аргументом должно включать тождественную функцию.

Рассмотрим пример кодирования рассмотренного в предыдущем разделе математического выражения

$$y = e^{-\alpha x_1} \cos(x_2 + \beta) + \sin|x_2|.$$

Для кодирования данного выражения используем множества функций:

$$F_0 = (f_{0,1} = x_1, f_{0,2} = x_2, f_{0,3} = \alpha, f_{0,4} = \beta),$$

$$F_1 = (f_{1,1}(z) = z, f_{1,2}(z) = -z, f_{1,3}(z) = e^z, f_{1,4}(z) = |z|, f_{1,5}(z) = \sin(z), f_{1,6}(z) = \cos(z)),$$

$$F_2 = (f_{2,1}(z_1, z_2) = z_1 + z_2, f_{2,2}(z_1, z_2) = z_1 \cdot z_2).$$

В отличие от множеств функций из предыдущего раздела здесь во множество функций с одним аргументом введена тождественная функция $f_{1,1}(z) = z$.

Запишем кодируемое математическое выражение с помощью элементов рассматриваемых функций

$$y = f_{2,1}(f_{2,2}(f_{1,3}(f_{2,2}(f_{0,1}, f_{1,2}(f_{0,3}))), f_{1,6}(f_{2,1}(f_{0,2}, f_{0,4}))), f_{1,5}(f_{1,4}(f_{0,2}))).$$

Данная запись математического выражения называется программной записью. Данное название объясняется легким представлением этой записи в виде кода программы любого языка программирования высокого уровня при условии реализации всех элементов множеств в виде подпрограмм-функций.

В сетевом операторе все функции с одним или двумя аргументами называют унарными или бинарными операциями.

Для построения графа сетевого оператора необходимо преобразовать программную запись в графическую. Графическая запись математического выражения обладает следующими свойствами:

- а)) первой функцией записи всегда является бинарная операция,
- б) аргументами унарных операций являются функции без аргументов или бинарные операции,
- в) аргументами бинарных операций являются унарные операции,
- г) аргументами одной бинарной операции не могут быть унарные операции, аргументом которых является одна и та же функция без аргументов.

Любую программную запись можно преобразовать в графическую запись с помощью введения дополнительных тождественных унарных операций и дополнительных бинарных операций с единичным элементом в качестве второго аргумента

$$f_{2,1}(z,0) = z + 0 = z.$$

Здесь 0 - единичный элемент бинарной операции сложения.

Отметим нарушения условий а) – г) в полученной для примера математического выражения программной записи

$$y = \underbrace{f_{2,1}(f_{1,2}(f_{2,2}(f_{1,3}(\underbrace{f_{2,2}(f_{0,1}, f_{1,2}(f_{0,3}))}_{\text{в}})), f_{1,6}(\underbrace{f_{2,1}(f_{0,2}, f_{0,4}))}_{\text{в}}))}_{\text{в}}, \underbrace{f_{1,5}(f_{1,4}(f_{0,2}))}_{\text{б}}}.$$

В программной записи обнаружено пять нарушений условий графической записи: четыре раза встречаем нарушения условия в) и один раз нарушение условия б).

Для устранения нарушений используем следующие соотношения:

$$f_{2,i}(f_{2,j}(\dots)) = f_{2,i}(f_{1,1}(f_{2,j}(\dots))) , \quad (1.49)$$

$$f_{2,i}(f_{0,j}(\dots)) = f_{2,i}(f_{1,1}(f_{0,j}(\dots))) , \quad (1.50)$$

$$f_{1,i}(f_{1,j}(\dots)) = f_{1,i}(f_{2,1}(f_{1,j}(\dots), 0)) . \quad (1.51)$$

В левых частях соотношений (1.49) – (1.51) элементы программной записи с нарушениями свойств в), б). В правых частях соотношений (1.49) – (1.51) аналогичные элементы графической записи без нарушения свойств в), б).

Используем соотношения (1.41) – (1.43) для устранения нарушений в программной записи. Получаем графическую запись математического выражения

$$y = f_{2,1}(f_{1,2}(f_{2,2}(f_{1,3}(f_{2,2}(f_{1,1}(f_{0,1}), f_{1,2}(f_{0,3}))), f_{1,6}(f_{2,1}(f_{1,1}(f_{0,2}), f_{1,1}(f_{0,4}))))), f_{1,5}(f_{2,1}(f_{1,4}(f_{0,2}), 0)))$$

По графической записи строим граф сетевого оператора. Бинарные операции соответствуют узлам графа, функции без аргументов соответствуют узлам-источникам графа, унарные операции соответствуют дугам графа.

Правила построения графа сетевого оператора по графической записи математического выражения приведены на рис. 1.2.

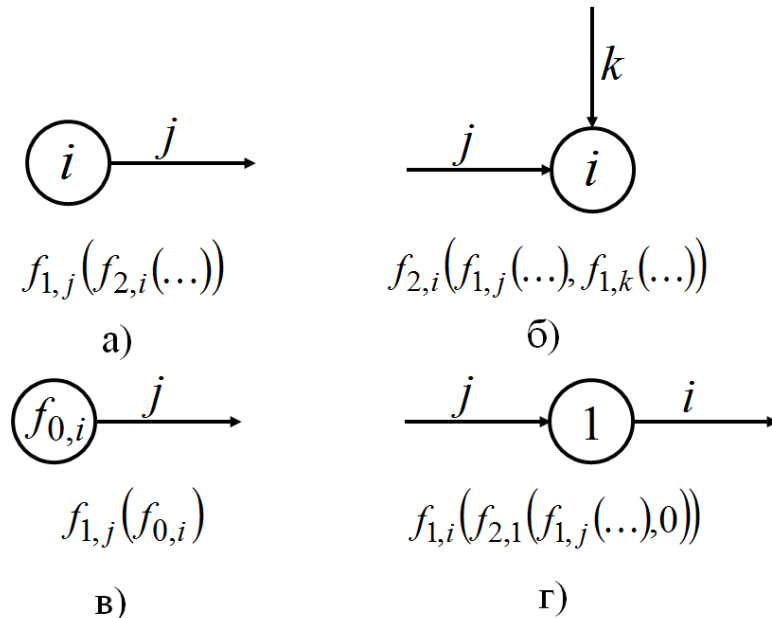


Рис. 1.2. Правила построения графа сетевого оператора

Используем правила для построения графа полученной графической записи рассматриваемого математического выражения. Полученный граф сетевого оператора приведен на рис. 1.3.

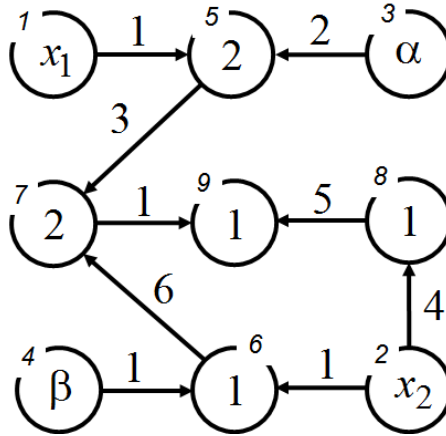


Рис. 1.3. Граф сетевого оператора математического выражения

На рис. 1.3 в узлах графа приведены номера бинарных операций. В узлах-источниках графа приведены используемые в математическом выражении функции без аргументов. На дугах графа указаны номера унарных операций.

В верхних частях узлов графа на рис. 1.3 приведены их номера. Порядок номеров узлов установлен в соответствие с топологической сортировкой. Номер узла на конце дуги должен быть больше номера узла у начала дуги. Для ориентированных графов без циклов топологическая сортировка номеров узлов всегда возможна.

Матрица смежности полученного графа сетевого оператора имеет вид:

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Номер строки матрицы смежности соответствует номеру узла графа. Вследствие топологической сортировки номеров узлов матрица смежности имеет

верхний треугольный вид. Единицы матрицы смежности соответствуют дугам графа. Единица в строке i в столбце j соответствует дуге из узла i в узел j .

Для построения матрицы сетевого оператора используют матрицу смежности графа сетевого оператора. В матрице смежности в местах расположения единиц указывают номера связанных с дугами унарных операций. На диагонали матрицы смежности указывают номера бинарных операций. В результате получаем матрицу сетевого оператора

$$\Psi = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 6 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 5 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Матрица сетевого оператора имеет верхний треугольный вид. Нулевые столбцы матрицы указывают на узлы источники графа. Номера столбца соответствует номеру элемента множества функций без аргументов F_0 . Матрица сетевого оператора является кодом записи математического выражения. Матрицы сетевого оператора вместе с множествами функций без аргументов F_0 , с одним аргументом F_1 и с двумя аргументами F_2 достаточно для вычисления закодированного ею математического выражения.

Опишем процедуру вычисления математического выражения по матрице сетевого оператора размерностью $K \times K$

$$\Psi = [\psi_{i,j}], \quad i, j = \overline{1, K}. \quad (1.52)$$

Для вычислений необходим дополнительный вектор узлов размерностью K для хранения промежуточных результатов вычислений.

Инициализируем вектор узлов. Начальные значения компонент вектора узлов равны значению функций без аргументов или единичным элементам бинарных операций

$$\mathbf{z}^{(0)} = [z_1^{(0)} \dots z_K^{(0)}]^T, \quad (1.53)$$

$$z_i^{(0)} = \begin{cases} f_{0,i}, & \text{если } \psi_{i,i} = 0 \\ e_{\psi_{i,i}} & \text{- иначе} \end{cases}, \quad i = \overline{1, K}, \quad (1.54)$$

где e_j - единичный элемент бинарной операции $f_{2,j}(z_1, z_2)$,

$$f_{2,j}(e_j, z_2) = z_2, \quad f_{2,j}(z_1, e_j) = z_1, \quad j = \overline{1, |\mathbb{F}_2|}. \quad (1.55)$$

Проходим последовательно все строки матрицы сетевого оператора. Изменяем значения компонент вектора узлов по формулам

$$z_j^{(i)} = \begin{cases} f_{2, \psi_{j,j}}(z_j^{(i-1)}, f_{1, \psi_{i,j}}(z_i^{(i-1)})), & \text{если } \psi_{i,j} \neq 0 \\ z_j^{(i-1)} & \text{- иначе} \end{cases}, \quad i = \overline{1, K-1}, \quad j = \overline{i+1, K}. \quad (1.56)$$

Рассмотрим вычисление рассмотренного примера математического выражения по полученной матрице сетевого оператора.

Инициализируем вектор узлов

$$\mathbf{z}^{(0)} = [x_1 \ x_2 \ \alpha \ \beta \ 1 \ 0 \ 1 \ 0 \ 0]^T.$$

Первые четыре компоненты вектора узлов совпадают со значениями соответствующих элементов множества F_0 функций без аргументов. Остальные значения компонент совпадают с единичными элементами соответствующих бинарных операций. Единичным элементом бинарной операции сложения $f_{2,1}(z_1, z_2) = z_1 + z_2$ является $e_1 = 0$. Единичным элементом бинарной операции умножения $f_{2,2}(z_1, z_2) = z_1 \cdot z_2$ является $e_2 = 1$.

В соответствие с соотношением (1.40) Проходим матрицу Ψ по первой строке и получаем

$$\psi_{1,5} = 1, \quad \psi_{5,5} = 2, \quad z_5^{(1)} = f_{2,2}(z_5^{(0)}, f_{1,1}(z_1^{(0)})) = 1 \cdot x = x;$$

$$z_j^{(1)} = z_j^{(0)}, \quad j \neq 5.$$

После прохода первой строки вектор узлов имеет вид

$$\mathbf{z}^{(1)} = [x_1 \ x_2 \ \alpha \ \beta \ x_1 \ 0 \ 1 \ 0 \ 0]^T.$$

Проходим следующие строки

$$\Psi_{2,6} = 1, \Psi_{6,6} = 1, z_6^{(2)} = f_{2,1}(z_6^{(1)}, f_{1,1}(z_2^{(1)})) = 0 + x_2 = x_2;$$

$$\Psi_{2,8} = 4, \Psi_{8,8} = 1, z_8^{(2)} = f_{2,1}(z_8^{(1)}, f_{1,4}(z_2^{(1)})) = 0 + |x_2| = |x_2|;$$

$$z_j^{(2)} = z_j^{(1)}, j \neq 6, j \neq 8.$$

$$\Psi_{3,5} = 2, \Psi_{5,5} = 2, z_5^{(3)} = f_{2,2}(z_5^{(2)}, f_{1,2}(z_3^{(2)})) = x_1 \cdot (-\alpha) = -x_1\alpha;$$

$$z_j^{(3)} = z_j^{(2)}, j \neq 5.$$

$$\Psi_{4,6} = 1, \Psi_{6,6} = 1, z_6^{(4)} = f_{2,1}(z_6^{(3)}, f_{1,1}(z_4^{(3)})) = x_2 + \beta;$$

$$z_j^{(4)} = z_j^{(3)}, j \neq 6.$$

$$\Psi_{5,7} = 3, \Psi_{7,7} = 2, z_7^{(5)} = f_{2,2}(z_7^{(4)}, f_{1,3}(z_5^{(4)})) = 1 \cdot e^{-x_1\alpha} = e^{-x_1\alpha};$$

$$z_j^{(5)} = z_j^{(4)}, j \neq 7.$$

$$\Psi_{6,7} = 6, \Psi_{7,7} = 2, z_7^{(6)} = f_{2,2}(z_7^{(5)}, f_{1,6}(z_6^{(5)})) = e^{-x_1\alpha} \cos(x_2 + \beta);$$

$$z_j^{(6)} = z_j^{(5)}, j \neq 7.$$

$$\Psi_{7,9} = 1, \Psi_{9,9} = 1, z_9^{(7)} = f_{2,1}(z_9^{(6)}, f_{1,1}(z_7^{(6)})) = 0 + e^{-x_1\alpha} \cos(x_2 + \beta);$$

$$z_j^{(7)} = z_j^{(6)}, j \neq 9.$$

$$\Psi_{8,9} = 5, \Psi_{9,9} = 1, z_9^{(8)} = f_{2,1}(z_9^{(7)}, f_{1,5}(z_7^{(5)})) = e^{-x_1\alpha} \cos(x_2 + \beta) + \sin|x_2|;$$

$$z_j^{(8)} = z_j^{(7)}, j \neq 9.$$

В итоге вектор узлов имеет следующие значения компонент:

$$\mathbf{z} = \begin{bmatrix} x_1 \\ x_2 \\ \alpha \\ \beta \\ -x_1\alpha \\ x_2 + \beta \\ e^{-x_1\alpha} \cos(x_2 + \beta) \\ |x_2| \\ e^{-x_1\alpha} \cos(x_2 + \beta) + \sin|x_2| \end{bmatrix}.$$

Результат вычисления математического выражения соответствует значению последней компоненты вектора узлов.

С помощью метода сетевого оператора было решено большое количество задач синтеза систем управления. При поиске оптимального решения метод сетевого оператора использует принцип малых вариаций базисного решения. Согласно этому принципу исследователь задает одно базисное решение. Поиск оптимального решения осуществляют на множестве малых вариаций базисного решения. Само базисное решение в процессе поиска меняют на наилучшее найденное к этому моменту решение.

Использование принципа малых вариаций приводит к применению процедуры построения матрицы сетевого оператора по математическому выражению только для одного базисного решения. В процессе поиска используют малые вариации и не рекурсивное соотношение (1.48) для вычисления математического выражения.

1.7. Вариационный генетический алгоритм для решения задачи поиска оптимальных траекторий

Рассмотрим метод решения задачи поиска оптимальных пространственных траекторий (1.1) – (1.8) на основе вариационного генетического алгоритма.

Для заданного множества точек (1.1) определим упорядоченное множество индексов

$$I(P) = (i_1, \dots, i_M), \quad (1.57)$$

Пусть первоначально каждый индекс равен своему порядковому номеру во множестве (1.48)

$$i_j = j, \quad j = 1, \dots, M. \quad (1.58)$$

Пусть количество точек, которые должен пройти каждый из квадрокоптеров, одинаково для всех объектов управления

$$K_1 = K_2 = \dots = K_{N-1} = d, \quad (1.59)$$

где

$$d = \left\lfloor \frac{M}{N} \right\rfloor. \quad (1.60)$$

Из-за округления (1.51) мощность K_N подмножества точек для последнего квадрокоптера определяем из соотношения

$$K_N = M - d(N-1). \quad (1.61)$$

Определим базисный набор подмножеств или базисное решение

$$\begin{aligned} \tilde{P}_1 &= (\mathbf{p}^{i_1}, \dots, \mathbf{p}^{i_{K_1}}), \\ \tilde{P}_2 &= (\mathbf{p}^{i_{K_1+1}}, \dots, \mathbf{p}^{i_{K_2}}), \\ &\vdots \\ \tilde{P}_N &= (\mathbf{p}^{i_{K_{N-1}+1}}, \dots, \mathbf{p}^{i_{K_N}}), \end{aligned} \quad (1.62)$$

где $\tilde{P}_1 \cup \dots \cup \tilde{P}_N = P$.

Согласно принципу малых вариаций базисного решения [71] введем вектор малых вариаций

$$\mathbf{w} = [w_1 \ w_2]^T, \quad (1.63)$$

где w_1 - номер позиции во множестве индексов (1.48), куда будет вставлен номер новой позиции, w_2 - номер позиции во множестве индексов (1.48), откуда будет перемещен номер в позицию w_1 .

Действие вектора малых вариаций (1.54) на множество индексов (1.48), определяющих базисное решение (1.53), приводит к следующим изменениям множества индексов, а, следовательно, и возможного решения

$$\mathbf{w} \circ I(P) = [w_1 \ w_2]^T \circ I(P) =$$

$$= (\overbrace{i_1, \dots, i_{w_1}}^{w_1}, i_{w_1+1}, \dots, i_{w_2-1}, i_{w_2+1}, \dots, i_M). \quad (1.64)$$

Вариационный генетический алгоритм работает с упорядоченными множествами векторов вариаций. Каждое множество векторов вариаций содержит определенное количество векторов малых вариаций. Упорядоченное множество векторов вариаций действует последовательно на множество индексов базисного решения и преобразует базисное решение в одно из возможных решений задачи. Основные генетические операции, скрещивания и мутации выполняются на упорядоченных множествах векторов вариаций.

Начальное множество возможных решений состоит из базисного решения (1.48), (1.53) и набора упорядоченных множеств векторов малых вариаций

$$W = \{W_1, \dots, W_H\}, \quad (1.65)$$

где W_i - упорядоченное множество векторов вариаций,

$$W_i = (\mathbf{w}^{i,1}, \dots, \mathbf{w}^{i,L}), \quad (1.66)$$

$$\mathbf{w}^{i,j} = [w_1^{i,j} \ w_2^{i,j}]^T, \quad i=1, \dots, H, \quad j=1, \dots, L. \quad (1.67)$$

Каждое упорядоченное множество векторов малых вариаций определяет новое возможное решение после воздействия на базисное решение

$$I_\alpha(P) = \mathbf{w}^{\alpha,1} \circ \dots \circ \mathbf{w}^{\alpha,L} \circ I(P). \quad (1.68)$$

Для выполнения операции скрещивания отбираем из начального множества два упорядоченных множества векторов вариаций

$$W_\alpha = [\mathbf{w}^{\alpha,1} \ \dots \ \mathbf{w}^{\alpha,L}]^T$$

и

$$W_\beta = [\mathbf{w}^{\beta,1} \ \dots \ \mathbf{w}^{\beta,L}]^T,$$

случайно выбираем точку скрещивания $\gamma \in \{1, \dots, L\}$ и обмениваем между собой в отобранных множествах элементы после точки скрещивания. В результате получаем два новых упорядоченных множества векторов вариаций, которые определяют два новых возможных решения

$$W_{H+1} = (\mathbf{w}^{\alpha,1}, \dots, \mathbf{w}^{\alpha,\gamma-1}, \mathbf{w}^{\beta,\gamma}, \dots, \mathbf{w}^{\beta,L}), \quad (1.69)$$

$$W_{H+2} = (\mathbf{w}^{\beta,1}, \dots, \mathbf{w}^{\beta,\gamma-1}, \mathbf{w}^{\alpha,\gamma}, \dots, \mathbf{w}^{\alpha,L}). \quad (1.70)$$

Для выполнения операции мутации для каждого из новых упорядоченных множеств, которые определяют новые возможные решения W_{H+1} и W_{H+2} , находим случайно точку мутации $\mu_i \in \{1, \dots, L\}$, $i=1,2$, и генерируем новые векторы малых вариаций $\mathbf{w}^{H+i,\mu_i} = [w_1^{H+i,\mu_i} \ w_2^{H+i,\mu_i}]^T$, $i=1,2$, которые устанавливаем в соответствующие случайно найденные позиции μ_i , $i=1,2$, во упорядоченные множество векторов вариаций.

Рассмотрим пример решения задачи группы коммивояжеров вариационным генетическим алгоритмом. Пусть четверем квадрокоптерам необходимо пройти 50 точек в трехмерном пространстве $\{x, y, z\}$. Координаты случайно сгенерированных точек в диапазоне $x \in [-100;100]$, $y \in [1;100]$, $z \in [-100;100]$ были следующими

$$\begin{aligned} P_0 = \{ & [-11,84 \ 37,60 \ 19,60]^T, [-66,88 \ 29,67 \ -9,23]^T, [1,30 \ 46,37 \ -26,40]^T, \\ & [-4,83 \ 35,97 \ -62,98]^T, [68,57 \ 5,78 \ -20,11]^T, [-41,46 \ 30,91 \ 90,59]^T, \\ & [52,00 \ 8,51 \ 32,77]^T, [22,09 \ 1,38 \ 59,63]^T, [78,92 \ 21,42 \ -43,00]^T, \\ & [55,76 \ 1,44 \ 42,62]^T, [-50,64 \ 4,49 \ -48,32]^T, [56,15 \ 7,16 \ 56,44]^T, \\ & [0,19 \ 26,25 \ -69,31]^T, [-28,15 \ 44,17 \ 16,04]^T, [14,89 \ 37,06 \ 54,78]^T, \\ & [-61,36 \ 34,64 \ -42,95]^T, [77,38 \ 13,93 \ 53,61]^T, [82,14 \ 4,18 \ -23,59]^T, \\ & [-38,31 \ 49,90 \ 18,68]^T, [49,22 \ 9,35 \ 63,24]^T, [-59,35 \ 34,12 \ -18,83]^T, \\ & [30,13 \ 20,19 \ 37,67]^T, [42,65 \ 29,27 \ -33,16]^T, [-6,71 \ 30,16 \ 61,35]^T, \\ & [-54,62 \ 44,83 \ 72,85]^T, [-10,79 \ 9,98 \ 86,83]^T, [9,73 \ 32,13 \ 62,58]^T, \\ & [55,26 \ 17,85 \ 72,49]^T, [-25,83 \ 43,10 \ -36,93]^T, [-70,29 \ 20,89 \ 86,46]^T, \\ & [-78,41 \ 35,89 \ -85,23]^T, [-34,29 \ 8,16 \ -58,45]^T, [-65,45 \ 26,00 \ -46,78]^T, \\ & [-1,74 \ 5,99 \ 10,11]^T, [-64,85 \ 23,08 \ 86,50]^T, [44,47 \ 37,58 \ -92,08]^T, \\ & [79,14 \ 41,70 \ -61,59]^T, [-81,35 \ 6,83 \ -9,91]^T, [-40,53 \ 12,18 \ -52,21]^T, \\ & [-0,16 \ 38,58 \ -7,67]^T, [-85,97 \ 43,99 \ -95,66]^T, [-4,92 \ 24,06 \ 97,69]^T, \\ & [-31,71 \ 34,32 \ -51,31]^T, [-57,90 \ 40,27 \ -87,43]^T, [-97,84 \ 12,85 \ -54,55]^T, \\ & [-15,16 \ 14,24 \ 94,79]^T, [95,42 \ 32,51 \ -74,66]^T, [99,00 \ 35,73 \ -17,20]^T, \\ & [10,92 \ 30,13 \ -81,20]^T, [43,11 \ 20,02 \ -3,58]^T\}. \end{aligned}$$

Проекция траекторий движения квадрокоптеров на горизонтальную плоскость приведены на рис. 1.4

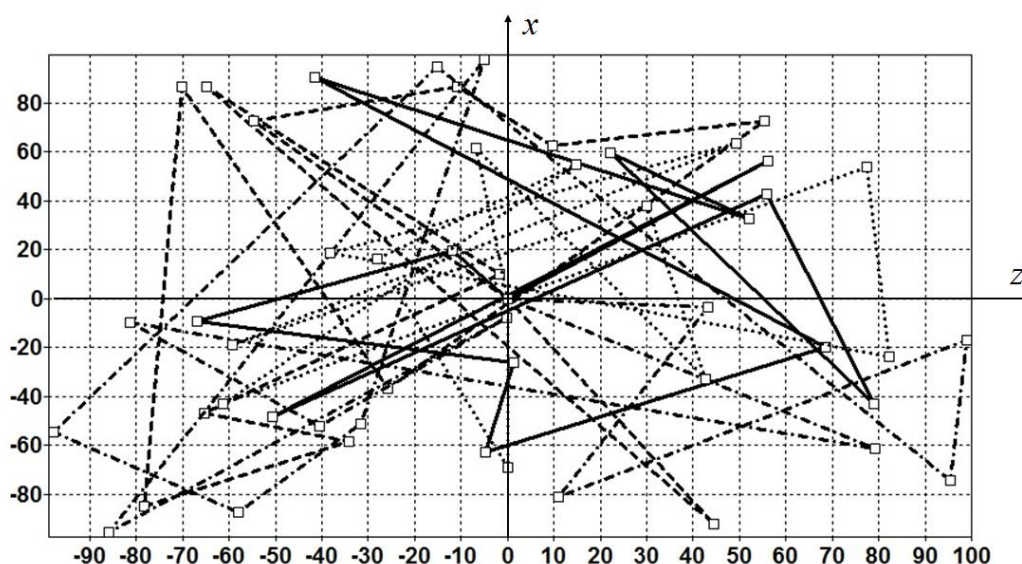


Рис. 1.4. Проекция неоптимальных траекторий

Длины неоптимальных путей каждого квадрокоптера составили соответственно: для первого квадрокоптера 1198,2, для второго 1131,1, для третьего 1294,2 и для четвертого 1667,2. Все квадрокоптеры стартовали из точки $[0 \ 0 \ 0]^T$ и туда возвращались. Неоптимальное движение квадрокоптеров дало значение суммарной длины

$$5490,7=1198,2+1131,1+1294,2+1667,2.$$

Для нахождения оптимальных траекторий квадрокоптеров использовали вариационный генетический алгоритм со следующими параметрами: мощность исходного множества возможных решений 1024, число поколений 512, количество скрещиваний в каждом поколении 256, число вариаций в одном возможном решении 12, число поколений перед сменой базисного решения 16, вероятность мутации 0,7.

В результате применения вариационного генетического программирования получили следующий порядок точек

$$I(P) = (1, 14, 19, 2, 21, 16, 33, 11, 39, 32, 43, 29, 34, 40, 3, 23, 48, 17, 28, 20, \\ 12, 10, 7, 50, 22, 8, 15, 27, 24, 26, 46, 42, 6, 35, 30, 25, 38, 45, 31, 41, 44, 4, \\ 13, 49, 36, 47, 37, 9, 18, 5).$$

Проекция оптимальных траекторий квадрокоптеров на горизонтальную плоскость приведены на рис. 1.5.

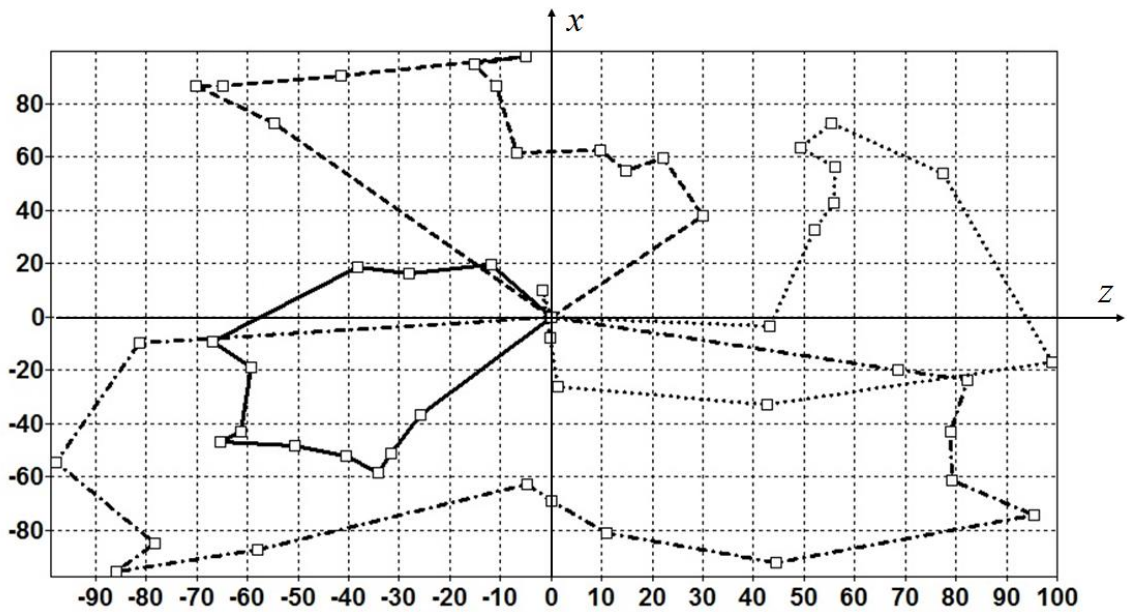


Рис. 1.5. Проекция оптимальных траекторий

Длины путей квадрокоптеров имели следующие значения: у первого квадрокоптера 322,8, у второго 418,4, у третьего 405,4, у четвертого 558,0. Суммарная длина пути составила величину

$$1704,6=322,8+418,4+405,4+558,0.$$

В результате использования вариационного генетического алгоритма удалось получить сокращение суммарной длины пути более, чем в три раза.

1.8. Выводы к главе 1

1. Сформулирована задача поиска оптимальных пространственных траекторий для группы движущихся объектов в форме задачи группы коммивояжеров, в которой необходимо отсортировать множество точек так, чтобы суммарная длина всех траекторий движения по точкам каждого коммивояжера была минимальной
2. Сформулирована задача синтеза системы управления. Задача включает математическую модель объекта управления в виде системы обыкновенных дифференциальных уравнений, множество начальных условий, терминальное многообразие и функционал качества. Решением задачи синтеза является многомерная функция многих переменных, которая описывает функциональную зависимость управления от значений компонент вектора состояния объекта управления.
3. Сформулирована задача численного синтеза системы управления. Задача является результатом уточнения формулировки задачи синтеза управления. В задаче численного синтеза управления множество начальных значений заменено конечным множеством точек начальных условий. Терминальные условия заменены дополнительным функционалом, описывающим отклонение от цели или норму отклонения от заданного многообразия. Переопределены функционалы качества управления в форме сумм значений функционалов, вычисляемых для всех заданных точек начальных условий.
4. Приведен обзор аналитических методов решения задачи синтеза управления. Показано, что аналитические методы имеют ограниченную область применения и в практических задачах используется в основном метод технического синтеза.
5. Приведен обзор методов символьной регрессии, которые могут быть применены к решению задачи численного синтеза системы управления. Подробно описан метод аналитического программирования и метод сетевого оператора. Представлены формы кодирования математических

алгоритмов. Указано, что метод аналитического программирования использует любые функции с любым количеством аргументов, а преимуществом метода сетевого оператора является принцип малых вариаций базисного решения.

6. Приведен пример решения задачи группы коммивояжеров для пятидесяти точек в трехмерном пространстве и четырех квадрокоптеров вариационным генетическим алгоритмом. Найденное оптимальное решение так перенумеровало точки, что длина суммарной траектории уменьшилась более, чем в три раза.

2. МЕТОД ВАРИАЦИОННОГО АНАЛИТИЧЕСКОГО ПРОГРАММИРОВАНИЯ ДЛЯ СИНТЕЗА СИСТЕМЫ ОПТИМАЛЬНОГО УПРАВЛЕНИЯ

Метод сетевого оператора успешно применяют к решению задачи синтеза управления. Удачное использование метода сетевого оператора в решении сложной задачи синтеза управления связано с применением при поиске решения принципа малых вариаций базисного решения. Согласно данному принципу исследователь задает на основе анализа задачи одно базисное решение в форме закодированного матрицей сетевого оператора математического выражения, а эволюционный алгоритм поиска ищет оптимальное решение на пространстве малых вариаций заданного базисного решения. В сложных задачах синтеза управления удачно построенное базисное решение повышает эффективность поиска и сокращает пространство поиска решения. Базисное решение может быть построено на основе усовершенствования найденных алгоритмом поиска решения. Принцип малых вариаций базисного решения объединяет усилия исследователя и вычислительной машины. С одной стороны исследователь анализирует полученные машиной решения и уточняет его. С другой стороны уточненное базисное решение позволяет вычислительной машине найти быстрее более лучшее оптимальное решение.

Применение принципа малых вариаций базисного решения к методу аналитического программирования привело к созданию нового метода символьной регрессии, методу вариационного аналитического программирования. В задаче синтеза системы управления необходимо найти функцию, которая описывает зависимость управления от координат пространства состояний. Найденная функция должна обеспечить достижение объектом цели управления из разных начальных состояний с оптимальным значением критерия качества.

Наиболее успешно решает задачи синтеза управления метод сетевого оператора [15–18]. Недостатком метода сетевого оператора является ограниченный набор используемых при построении математического выражения

функций. При решении сложных задач управления для эффективной работы алгоритма исследователь должен подобрать хорошее базисное решение. Для сложных систем управления робототехническими объектами приходится использовать два сетевых оператора с арифметическими и логическими функциями [61,62]. Метод аналитического программирования использует функции с любым количеством аргументов. Применение принципа малых вариаций базисного решения для метода аналитического программирования с целью решения задач синтеза управления. Привело к созданию нового метода символьной регрессии, метода вариационного аналитического программирования [20–31, 67, 68].

2.1 Вариационное аналитическое программирование

Метод использует упорядоченное множество функций

$$F = \left(f_1 = f_{0,1}, f_2 = f_{0,2}, \dots, f_{m_0} = f_{0,m_0}, f_{m_0+1}(z) = f_{1,1}(z), \dots, f_{m_0+m_1}(z) = f_{1,m_1}(z), \right. \\ \left. f_{m_0+m_1+1}(z_1, z_2) = f_{2,1}(z_1, z_2), \dots, f_{m_0+\dots+m_n}(z_1, \dots, z_n) = f_{n,m_n}(z_1, \dots, z_n) \right), \quad (2.1)$$

где $f_{i,j}(z_1, \dots, z_i)$ - функция под номером j , $j = 1, \dots, m_i$, с количеством аргументов i , $i = 0, \dots, n$.

Отдельно используем множество функций без аргументов. Данные функции в математических выражениях являются параметрами или переменными

$$F_0 = (f_{0,1}, \dots, f_{0,m_0}) = (x_1, \dots, x_N, q_1, \dots, q_p). \quad (2.2)$$

Для получения математического выражения из записи кода необходимо знать количества элементов в каждом из множеств функций: m_0, \dots, m_n , и количества используемых переменных и параметров N и p .

Для описания кодов векторных математических выражений используем один набор целых чисел с определенным количеством элементов для каждой компоненты векторного выражения. Пусть вектор математических выражений имеет m компонент. Выделим под каждую компоненту вектора L позиций в коде записи. Код математического выражения каждой компоненты i векторного

выражения содержит $k_i \leq L$ элементов, $i = \overline{1, m}$. Не используемым в формировании кода элементам задаем нулевые значения

$$C = \left(\underbrace{c_1, \dots, c_{k_1}, 0, \dots, 0, \dots}_L, \dots \right), \quad (2.3)$$

где L - число позиций для кода одной компоненты векторного выражения, k_i - длина кода компоненты i векторного выражения, $i = \overline{1, m}$.

В коде записи векторного выражения значения для элементов выполняются условия

$$c_j = 0, \text{ если } L(i-1) + k_i < j \leq Li, \quad i = \overline{1, m}. \quad (2.4)$$

Для расшифровки кода векторного выражения необходимо вместе с величинами m_i , $i = \overline{1, n}$, N и p дополнительно знать число позиций L и длины кодов компонент k_i , $i = \overline{1, m}$.

Для определения корректности записи кода математического выражения используем индекс элемента. Пусть в записи кода математического выражения элемент $c_j \neq 0$. Для векторного математического выражения определим него номер компоненты i из соотношения

$$i = \left\lfloor \frac{j-1}{L} \right\rfloor + 1. \quad (2.5)$$

Для корректной записи для индекса элемента $c_j \neq 0$ необходимо выполнения условий

$$T(j) > 0, \quad j \neq k_i, \quad (2.6)$$

$$T(k_i + L(i-1)) = 0. \quad (2.7)$$

где $T(j)$ индекс элемента j записи кода векторного математического выражения.

Для вычисления индекса $T(j)$ элемента j при условии $c_j \neq 0$ используем соотношение

$$T(j) = 1 - (j - L\beta) + \sum_{k=L\beta+1}^j i_k, \quad (2.8)$$

где

$$i_k = \begin{cases} 0, & \text{если } 0 \leq c_k \leq m_0 \\ \alpha, & \text{если } \sum_{r=0}^{\alpha-1} m_r \leq c_k \leq \sum_{r=0}^{\alpha} m_r, \alpha = \overline{1, m_n} \end{cases}, \quad (2.9)$$

$$\beta = \left\lfloor \frac{j-1}{L} \right\rfloor. \quad (2.10)$$

Индекс $T(j)$ элемента j указывает на минимальное число недостающих справа элементов. Невыполнение условий (2.6) или (2.7) указывает на неправильность записи математического выражения.

Пусть запись кода (2.6) (2.3) определяет пространства записей математических выражений. Пространство всех вариантов возможных записей длины K для M значений кодов содержит M^K элементов.

Малой вариацией кода аналитического программирования является изменение значения кода элемента $c_i \in \{1, \dots, M\}$. Одной малой вариации достаточно для получения любой записи пространства из одной заданной записи за конечное число вариаций.

Для описания малой вариации используем вектор из двух компонент

$$\mathbf{w} = [w_1 \ w_2]^T, \quad (2.11)$$

где w_1 - номер позиции изменяемого кода элемента записи, w_2 - новое значение кода элемента.

Для выполнения вариации необходимо знать: количество элементов в векторном выражении m , количество переменных N , количество параметров p , количества используемых функций определенной аргументности $m_0 = N + p$, m_1, \dots, m_n , количество выделенных для каждой компоненты позиций L и количество используемых в варьируемом коде позиций k_i , $i = \overline{1, m}$.

Номер позиции w_1 в векторе вариации (2.11) не должен превышать количество позиций в коде

$$1 \leq w_1 \leq Lm \quad (2.12)$$

Значение кода элемента w_2 не должно превышать количество элементов в объединенном множестве (2)

$$1 \leq w_2 \leq |F|, \quad (2.13)$$

где

$$|F| = N + p + \sum_{i=1}^n m_i. \quad (2.14)$$

При выполнении вариации возможно получение некорректных записей. Проверку корректности записи кода осуществляем по условиям (2.6), (2.7). Для обеспечения корректности записи в зависимости от полученного в результате вариации кода записи изменяем значения некоторых последних кодов. Рассмотрим выполнение малой вариации записи аналитического программирования подробнее.

Пусть заданы код записи $C = (c_1, \dots, c_{LM})$ и вектор малой вариации $\mathbf{w} = [w_1 \ w_2]^T$. В результате вариации получаем новое значение записи $\tilde{C} = (\tilde{c}_1, \dots, \tilde{c}_{LM})$ и новые значения количеств используемых позиций $\tilde{k}_i, i = \overline{1, m}$.

Определяем номер компоненты i векторного математического выражения

$$i = \left\lfloor \frac{w_1 - 1}{L} \right\rfloor + 1. \quad (2.15)$$

Определяем номер позиции j в коде записи. Номер позиции в коде компоненты i векторного выражения может попасть на неиспользуемую позицию или нулевой код элемента. Для исключения этого случая используем операцию вычисления остатка от деления

$$j = (w_1 - (i - 1)L - 1) \bmod k_i + L(i - 1) + 1. \quad (2.16)$$

По формуле (2.16) определяем варьируемую позицию j кода записи. Значение найденной позиции варьируемого кода не должно быть нулевым $c_j \neq 0$. Определяем арности элементов c_j и второй компоненты вектора вариаций w_2 .

Арность элемента указывает на количество аргументов кодируемой данным элементом функции. Обозначим $\gamma(a)$ арность кода a . Арность кода a определяем из соотношения

$$\gamma(a) = \begin{cases} 0, & \text{если } 0 < a \leq m_0 \\ k, & \text{если } \sum_{j=0}^{k-1} m_j < a \leq \sum_{j=0}^k m_j, k = \overline{1, m_n}. \end{cases} \quad (2.17)$$

После определения $\gamma(c_j)$ и $\gamma(w_2)$ возможны три случая.

Пусть арности кодов совпадают

$$\gamma(c_j) = \gamma(w_2). \quad (2.18)$$

В этом случае выполняем замену значения кода

$$\tilde{c}_j = w_2 \quad (2.19)$$

и завершаем вариацию.

Пусть арность кода заменяемого элемента превышает арность нового значения кода

$$\gamma(c_j) > \gamma(w_2). \quad (2.20)$$

В данном случае в коде записи для корректности необходимо обнулить значения части элементов.

Заменяем значение кода

$$\tilde{c}_j = w_2.$$

Определяем количество элементов в корректной записи кода. По формулам (2.8) - (2.9) определяем индексы $T(j)$ элемента \tilde{c}_j с новым значением w_2 и последующих за ним элементов $c_{j+1}, c_{j+2} \dots$

Находим элемент c_{j+v} с нулевым значением индекса $T(j+v) = 0$. После вариации кода элемент c_j принял новое значение с меньшей величиной арности (2.20). Условие указывает на нахождение элемента с нулевым индексом среди используемых позиций кода

$$j+v-L(i-1) < k_i. \quad (2.21)$$

Обнуляем все последующие за c_{j+v} элементы в коде записи компоненты i векторного выражения

$$\tilde{c}_{j+v+\alpha} = 0, \alpha = \overline{1, L(i-1) + k_i - j - v}. \quad (2.22)$$

Вариацию завершаем изменением числа используемых позиций в коде записи

$$\tilde{k}_i = j + v - L(i-1). \quad (2.23)$$

Пусть арность кода заменяемого элемента меньше арности кода нового значения

$$\gamma(c_j) < \gamma(w_2). \quad (2.24)$$

В данном случае необходимо добавить к коду записи некоторое число ненулевых позиций. Количество минимально добавляемых позиций определим из соотношения

$$\Delta = \gamma(w_2) - \gamma(c_j). \quad (2.25)$$

Всего для записи кода компоненты i в выделенных L позициях используется k_i позиций. Дополнительно можно использовать $L - k_i$ позиций. Для корректировки кода необходимо выполнения условия

$$\gamma(w_2) - \gamma(c_j) \leq L - k_i. \quad (2.26)$$

При нарушении условия (2.25)

$$\gamma(w_2) - \gamma(c_j) > L - k_i \quad (2.27)$$

вариацию кода не выполняем.

При выполнении условия (2.26) заменяем значение элемента c_j новым значением w_2 , $\tilde{c}_j = w_2$. Из-за условия (2.24) индекс последнего элемента в записи будет иметь ненулевое значение

$$T(L(i-1) + k_i) = \Delta \quad (2.28)$$

Для корректировки записи заменяем Δ нулевых значений кодами элементов с нулевой арностью. Для определения значений кодов используем остаток от деления на мощность множества элементов с нулевой арностью

$$\tilde{c}_{L(i-1)+k_i+\alpha} = \delta_\alpha, \alpha = \overline{1, \Delta}, \quad (2.29)$$

где

$$\delta_\alpha = (w_2 + \delta_{\alpha-1}) \bmod m_0 + 1, \alpha = \overline{1, \Delta}, \delta_0 = 0. \quad (2.30)$$

Завершаем вариацию изменением количества используемых позиций в записи кода

$$\tilde{k}_i = L(i-1) + k_i + \Delta. \quad (2.31)$$

Рассмотрим примеры выполнения вариаций. Пусть задано множество функций

$$F = (f_1 = x_1, f_2 = x_2, f_3 = \alpha, f_4 = \beta, f_5 = -z, f_6 = e^z, f_7 = |z|, \\ f_8 = \sin(z), f_9 = \cos(z), f_{10} = z_1 + z_2, f_{11} = z_1 \cdot z_2)$$

и задан код записи векторного математического выражения

$$\mathbf{y} = \begin{bmatrix} e^{-\alpha x_1} \cos(x_2 + \beta) \\ \sin|x_2| \end{bmatrix},$$

$$C = (11, 6, 11, 1, 5, 3, 9, 10, 2, 3, 0, 0, 8, 7, 2, 0, 0, 0, 0, 0, 0, 0, 0).$$

Для кода записи имеем следующие значения параметров: $L = 12$, $m = 2$, $N = 2$, $p = 2$, $m_0 = 4$, $m_1 = 5$, $m_2 = 2$, $k_1 = 10$, $k_2 = 3$.

Пусть определен вектор вариаций

$$\mathbf{w} = [21 \ 1]^T.$$

Определяем из (2.15) номер компоненты векторного выражения

$$i = \left\lfloor \frac{w_1 - 1}{L} \right\rfloor + 1 = \left\lfloor \frac{21 - 1}{12} \right\rfloor + 1 = 2.$$

Определяем из (2.16) позицию варьируемого элемента

$$j = (w_1 - (i-1)L - 1) \bmod k_i + L(i-1) + 1 = \\ (21 - 12(2-1) - 1) \bmod 3 + 12(2-1) + 1 = 15.$$

В коде записи $c_{15} = 2$. Определяем по формуле (2.17) арности элементов c_{11} и второй компоненты вектора вариаций $w_2 = 1$. Получаем

$$\gamma(c_{15}) = \gamma(2) = 0, \quad 0 < 2 \leq m_0 = 4,$$

$$\gamma(w_2) = \gamma(1) = 0, \quad 0 < 1 \leq m_0 = 4.$$

Арности элементов совпадают. Изменяем значение элемента c_{15} . Новое значение кода элемента равно значению второй компоненты вектора вариаций, $c_{15} = w_2 = 1$. Получаем новый код записи

$$C = (11, 6, 11, 1, 5, 3, 9, 10, 2, 3, 0, 0, 8, 7, 1, 0, 0, 0, 0, 0, 0, 0, 0).$$

Новый код соответствует векторному математическому выражению

$$\mathbf{y} = \begin{bmatrix} e^{-\alpha x_1} \cos(x_2 + \beta) \\ \sin|x_1| \end{bmatrix}.$$

Рассмотрим вектор вариаций

$$\mathbf{w} = [16 \ 6]^T.$$

Определяем по формуле (2.15) номер компоненты

$$i = \left\lfloor \frac{w_1 - 1}{L} \right\rfloor + 1 = \left\lfloor \frac{16 - 1}{12} \right\rfloor + 1 = 2.$$

Определяем из (2.16) позицию варьируемого элемента

$$j = (w_1 - (i - 1)L - 1) \bmod k_i + L(i - 1) + 1 = \\ (16 - 12(2 - 1) - 1) \bmod 3 + 12(2 - 1) + 1 = 13.$$

В коде записи $c_{13} = 8$. Определяем по формуле (2.17) арности элементов c_{13} и второй компоненты вектора вариаций $w_2 = 6$. Получаем

$$\gamma(c_{13}) = \gamma(8) = 1, \quad m_0 < 8 \leq m_0 + m_1, \quad 4 < 8 \leq 4 + 5,$$

$$\gamma(w_2) = \gamma(6) = 1, \quad m_0 < 6 \leq m_0 + m_1, \quad 4 < 6 \leq 4 + 5.$$

Арности элементов вновь совпадают. Изменяем значение элемента $c_{13} = 6$.

Получаем новый код записи

$$C = (11, 6, 11, 1, 5, 3, 9, 10, 2, 3, 0, 0, 6, 7, 1, 0, 0, 0, 0, 0, 0, 0, 0).$$

Новый код соответствует векторному математическому выражению

$$\mathbf{y} = \begin{bmatrix} e^{-\alpha x_1} \cos(x_2 + \beta) \\ e^{|x_1|} \end{bmatrix}.$$

Рассмотрим вектор вариаций

$$\mathbf{w} = [3 \ 6]^T.$$

Определяем по формуле (2.15) номер компоненты

$$i = \left\lfloor \frac{w_1 - 1}{L} \right\rfloor + 1 = \left\lfloor \frac{3 - 1}{12} \right\rfloor + 1 = 1.$$

Определяем из (2.16) позицию варьируемого элемента

$$j = (w_1 - (i - 1)L - 1) \bmod k_i + L(i - 1) + 1 = \\ (3 - 12(1 - 1) - 1) \bmod 10 + 12(1 - 1) + 1 = 3.$$

Определяем по формуле (2.17) арности элементов c_3 и второй компоненты вектора вариаций $w_2 = 6$. Получаем

$$\gamma(c_3) = \gamma(11) = 2, \quad m_0 + m_1 < 11 \leq m_0 + m_1 + m_2, \quad 4 + 5 < 11 \leq 4 + 5 + 2, \\ \gamma(w_2) = \gamma(6) = 1, \quad m_0 < 6 \leq m_0 + m_1, \quad 4 < 6 \leq 4 + 5.$$

Выполняется условие (2.20), арность заменяемого элемента больше арности нового значения кода

$$\gamma(c_3) > \gamma(w_2), \quad \gamma(11) > \gamma(6).$$

Заменяем значение элемента кода записи новым значением $c_3 = 6$.

Получаем новый код записи

$$C = (11, 6, 6, 1, 5, 3, 9, 10, 2, 3, 0, 0, 0, 6, 7, 1, 0, 0, 0, 0, 0, 0, 0, 0).$$

Находим подстроку кода подвыражения после замененного элемента c_3 .

Подстрока состоит из одного элемента

$$C(3) = (c_4).$$

Индекс первого элемента подстроки равен нулю

$$T(C(3))(1) = 1 + \gamma(c_4) - 1 = 1 + 0 - 1 = 0.$$

Удаляем найденную подстроку. Получаем новый код

$$C = (11, 6, 6, 5, 3, 9, 10, 2, 3, 0, 0, 0, 6, 7, 1, 0, 0, 0, 0, 0, 0, 0, 0).$$

Полученный код соответствует математическому выражению

$$\mathbf{y} = \begin{bmatrix} e^{e^{-\alpha}} \cos(x_2 + \beta) \\ e^{|x_1|} \end{bmatrix}.$$

Рассмотрим вектор вариаций

$$\mathbf{w} = [2 \ 10]^T.$$

Определяем по формуле (2.15) номер компоненты

$$i = \left\lfloor \frac{w_1 - 1}{L} \right\rfloor + 1 = \left\lfloor \frac{2 - 1}{12} \right\rfloor + 1 = 1.$$

Определяем из (2.16) позицию варьируемого элемента

$$j = (w_1 - (i - 1)L - 1) \bmod k_i + L(i - 1) + 1 = \\ (2 - 12(1 - 1) - 1) \bmod 9 + 12(1 - 1) + 1 = 2.$$

Определяем по формуле (2.17) арности элементов c_2 и второй компоненты вектора вариаций $w_2 = 10$. Получаем

$$\gamma(c_2) = \gamma(6) = 1, \quad m_0 < 6 \leq m_0 + m_1, \quad 4 < 6 \leq 4 + 5, \\ \gamma(w_2) = \gamma(10) = 2, \quad m_0 + m_1 < 10 \leq m_0 + m_1 + m_2, \quad 4 + 5 < 10 \leq 4 + 5 + 2.$$

Арность кода заменяемого элемента меньше арности кода нового значения.

В данном случае получаем условие (2.24)

$$\gamma(c_2) < \gamma(w_2), \quad \gamma(6) < \gamma(10).$$

По формуле (2.25) вычисляем минимально необходимое количество неиспользуемых в коде записи позиций

$$\Delta = \gamma(w_2) - \gamma(c_j) = 2 - 1 = 1.$$

Проверяем условие (2.26)

$$L - k_1 = 12 - 9 = 3 > 0,$$

Условие (2.26) выполняется. Вариация кода осуществима. Заменяем значение элемента $c_2 = 10$. Добавляем вместо неиспользуемых нулевых позиций элементы с нулевой арностью по формулам (2.30), (2.31)

$$\delta_1 = (w_2 + \delta_0) \bmod m_0 + 1 = (10 + 0) \bmod 4 + 1 = 3, \quad c_{10} = \delta_1 = 3,$$

В результате получаем новую запись кода

$$C = (11, 10, 6, 5, 3, 9, 10, 2, 3, 3, 0, 0, 6, 7, 1, 0, 0, 0, 0, 0, 0, 0, 0).$$

Код записи соответствует математическому выражению

$$\mathbf{y} = \begin{bmatrix} (e^{-\alpha} + \cos(x_2 + \beta))\beta \\ e^{|x_1|} \end{bmatrix}.$$

В результате действия набора векторов вариаций на код записи математического выражения получили код нового математического выражения.

Условно выполненные операции запишем в следующем виде:

$$\begin{bmatrix} (e^{-\alpha} + \cos(x_2 + \beta))\beta \\ e^{|x_1|} \end{bmatrix} = \begin{bmatrix} 21 \\ 1 \end{bmatrix} \circ \begin{bmatrix} 16 \\ 6 \end{bmatrix} \circ \begin{bmatrix} 3 \\ 6 \end{bmatrix} \circ \begin{bmatrix} 2 \\ 10 \end{bmatrix} \circ \begin{bmatrix} e^{-\alpha x_1} \cos(x_2 + \beta) \\ \sin|x_2| \end{bmatrix}.$$

Математическое выражение $\mathbf{y} = \begin{bmatrix} (e^{-\alpha} + \cos(x_2 + \beta))\beta \\ e^{|x_1|} \end{bmatrix}$ получено из выражения $y = \begin{bmatrix} e^{-\alpha x_1} \cos(x_2 + \beta) \\ \sin|x_2| \end{bmatrix}$ с помощью четырех малых вариаций или находится в окрестности исходного выражения на расстоянии четырех малых вариаций.

2.2. Генетический алгоритм на основе малых вариаций базисного решения

Генетический алгоритм для организации поиска математического выражения с помощью метода вариационного аналитического программирования выполняет все основные генетические операции на наборах векторов малых вариаций.

Рассмотрим основные шаги алгоритма.

Задаем базисное математическое выражение

$$\mathbf{y} = \mathbf{h}^0(\mathbf{x}, \mathbf{q}^0), \quad (2.32)$$

где $\mathbf{h}^0(\mathbf{x}, \mathbf{q}^0) = [h_1^0(\mathbf{x}, \mathbf{q}^0) \dots h_M^0(\mathbf{x}, \mathbf{q}^0)]^T$, $\mathbf{x} = [x_1 \dots x_N]^T$, $\mathbf{q}^0 = [q_1^0 \dots q_p^0]^T$.

Кодируем базисное решение кодом записи аналитического программирования

$$\mathbf{c}^0 = (c_1^0, \dots, c_{Lm}^0). \quad (2.33)$$

Задаем множество наборов векторов вариаций

$$\mathbf{W} = (\mathbf{W}^1, \dots, \mathbf{W}^H), \quad (2.34)$$

где

$$\mathbf{W}^i = (\mathbf{w}^{i,1}, \dots, \mathbf{w}^{i,r}), \quad (2.35)$$

$$\mathbf{w}^{i,j} = [w_1^{i,j} \ w_2^{i,j}]^T, \quad i = \overline{1, H}, \quad j = \overline{1, r}. \quad (2.36)$$

Генетические операции скрещивания и мутации осуществляем на наборах (2.35) векторов вариаций базисного решения.

Базисное значение вектора параметров кодируем в форме двоично-симметричного кода Грея

$$\mathbf{z}^0 = [z_1^0 \ \dots \ z_{p(c+d)}^0]^T, \quad (2.37)$$

где $z_i^0 \in \{0,1\}$, $i = \overline{1, p(c+d)}$, c - число бит для целой части числа, d - число бит для дробной части числа.

Каждое решение определяем с помощью набора заданной длины r из векторов вариаций.

Для определения базисного решения определяем набор из нулевых векторов вариаций

$$\mathbf{W}^0 = (\mathbf{w}^{0,1}, \dots, \mathbf{w}^{0,r}), \quad (2.38)$$

где $\mathbf{w}^{0,i} = [0 \ 0]^T$, $i = \overline{1, r}$.

Генерируем множество возможных решений в виде пар из кодов параметров и наборов векторов вариаций

$$\mathbf{G} = ((\mathbf{W}^0, \mathbf{z}^0), \dots, (\mathbf{W}^H, \mathbf{z}^H)) \quad (2.39)$$

Каждое возможное решение для вычисления значений функционалов преобразуем в записи кодов и вектор параметров

$$(\mathbf{W}^j, \mathbf{z}^j) \leftrightarrow (\mathbf{C}^j, \mathbf{q}^j), \quad (2.40)$$

где $\mathbf{C}^j = (c_1^j, \dots, c_{LM}^j)$,

$$\mathbf{C}^j = \mathbf{w}^{j,r} \circ \dots \circ \mathbf{w}^{j,1} \circ \mathbf{C}^0. \quad (2.41)$$

При выполнении операции скрещивания генетического алгоритма отбираем два возможных решения $(\mathbf{W}^{i_1}, \mathbf{z}^{i_1})$, $(\mathbf{W}^{i_2}, \mathbf{z}^{i_2})$.

Определяем две точки скрещивания k_1, k_2 , $1 \leq k_1 \leq r$, $1 \leq k_2 \leq p(c+d)$. Первая точка k_1 предназначена для скрещивания наборов векторов вариаций. Вторая точка k_2 предназначена для скрещивания кодов векторов параметров.

В результате выполнения операции скрещивания получаем четыре новых возможных решения $(\mathbf{W}^{H+i}, \mathbf{z}^{H+i})$, $i = \overline{1,4}$. Два новых решения получаем в результате скрещивания наборов векторов вариаций и кодов Грея для векторов параметров. Два других новых решения получаем в результате скрещивания кодов Грея только для векторов параметров

$$\mathbf{W}^{H+1} = (\mathbf{w}^{i_1,1}, \dots, \mathbf{w}^{i_1,k_1-1}, \mathbf{w}^{i_2,k_1}, \dots, \mathbf{w}^{i_2,r}), \quad (2.42)$$

$$\mathbf{W}^{H+2} = (\mathbf{w}^{i_1,2}, \dots, \mathbf{w}^{i_2,k_1-1}, \mathbf{w}^{i_1,k_1}, \dots, \mathbf{w}^{i_1,r}), \quad (2.43)$$

$$\mathbf{W}^{H+3} = \mathbf{W}^{i_1}, \quad (2.44)$$

$$\mathbf{W}^{H+4} = \mathbf{W}^{i_2}, \quad (2.45)$$

$$\mathbf{z}^{H+1} = \left[z_1^{i_1} \dots z_{k_2-1}^{i_1} z_{k_2}^{i_2} \dots z_{p(c+d)}^{i_2} \right]^T, \quad (2.46)$$

$$\mathbf{z}^{H+2} = \left[z_1^{i_2} \dots z_{k_2-1}^{i_2} z_{k_2}^{i_1} \dots z_{p(c+d)}^{i_1} \right]^T, \quad (2.47)$$

$$\mathbf{z}^{H+3} = \left[z_1^{i_1} \dots z_{k_2-1}^{i_1} z_{k_2}^{i_2} \dots z_{p(c+d)}^{i_2} \right]^T, \quad (2.48)$$

$$\mathbf{z}^{H+4} = \left[z_1^{i_2} \dots z_{k_2-1}^{i_2} z_{k_2}^{i_1} \dots z_{p(c+d)}^{i_1} \right]^T. \quad (2.49)$$

Выполняем операцию мутации для каждого нового возможного решения $(\mathbf{W}^{H+i}, \mathbf{z}^{H+i})$, $i = \overline{1,4}$. Случайно с заданной вероятностью мутации P_μ для каждого нового решения определяем точки мутации m_1, m_2 , $1 \leq m_1 \leq r$, $1 \leq m_2 \leq p(c+d)$. Генерируем новое значение вектора вариаций $\tilde{\mathbf{w}}^{H+i}$ и новое значение компоненты $\tilde{z}^{H+i} \in \{0,1\}$ вектора параметров. Помещаем сгенерированные значения в позиции m_1, m_2 новых возможных решений

$$\mathbf{W}^{H+i} = (\mathbf{w}^{H+i,1}, \dots, \mathbf{w}^{H+i,m_1-1}, \tilde{\mathbf{w}}^{H+i}, \mathbf{w}^{H+i,m_1+1}, \dots, \mathbf{w}^{H+i,r}), \quad (2.50)$$

$$\mathbf{z}^{H+i} = [z_1^{H+i} \dots z_{m_2-1}^{H+i} \tilde{z}^{H+i} z_{m_2+1}^{H+i} \dots z_{p(c+d)}^{H+i}]^T, \quad (2.51)$$

где $i = \overline{1,4}$.

Проверяем для каждого нового возможного решения значение функции приспособленности.

Остальные операции выполняем в соответствии с правилами выбранной для поиска решения разновидности генетического алгоритма.

2.3. Синтез системы управления методом вариационного аналитического программирования

Рассмотрим применение метода вариационного аналитического программирования к решению задачи синтеза управления для системы третьего порядка с фазовыми ограничениями [80].

Задана система уравнений

$$\dot{x}_1 = x_2, \quad (2.52)$$

$$\dot{x}_2 = x_3, \quad (2.53)$$

$$\dot{x}_3 = u. \quad (2.54)$$

Заданы терминальные условия

$$x_i(t_f) = 0, \quad i = 1, 2, 3. \quad (2.55)$$

Заданы фазовые ограничения

$$x_1 \geq 0. \quad (2.56)$$

Задана область начальных значений

$$x_1(0) = 1, \quad (2.57)$$

$$-1 \leq x_2(0) \leq 1, \quad (2.58)$$

$$-1 \leq x_3(0) \leq 1. \quad (2.59)$$

Задан функционал

$$J = \int_0^{t_f} \left(x_1 + \frac{u^2}{2} \right) dt \rightarrow \min . \quad (2.60)$$

Необходимо найти управление в виде

$$u = h(x_1, x_2, x_3).$$

Определим множество точек начальных условий

$$X_0 = \left\{ \mathbf{x}^{0,1} = [1 \ -1 \ -1]^T, \mathbf{x}^{0,2} = [1 \ -1 \ 1]^T, \mathbf{x}^{0,3} = [1 \ 1 \ -1]^T, \mathbf{x}^{0,4} = [1 \ 1 \ 1]^T \right\} . \quad (2.61)$$

Переопределим формулу для вычисления функционала с учетом множества начальных условий

$$J_1 = \sum_{i=1}^4 \left(\int_0^{t_f} \left(x_1 + \frac{u^2}{2} \right) dt \right)_{\mathbf{x}^{0,i}} \rightarrow \min . \quad (2.62)$$

Для решения задачи используем метод вариационного аналитического программирования.

Терминальные условия определим в виде дополнительного функционала

$$J_2 = \sum_{i=1}^4 |x_i(t_f)|_{\mathbf{x}^{0,i}} \rightarrow \min , \quad (2.63)$$

где

$$t_f = \begin{cases} t, & \text{если } \max \{ |x_i(t)|, i=1,2,3 \} < \varepsilon \\ t^+ & \text{иначе} \end{cases} , \quad (2.64)$$

t^+ - заданное предельное время моделирования, ε - заданная точность попадания в терминальное состояние.

Для учета фазовых ограничений (2.56) вводим функцию штрафа за их нарушение

$$p = \int_0^{t_f} \vartheta(-x_1) x_1^2 dt, \quad (2.65)$$

где $\vartheta(a)$ - функция Хэвисайда

$$\vartheta(a) = \begin{cases} 0, & \text{если } a < 0 \\ 1 & \text{иначе} \end{cases} .$$

С учетом функции штрафа (2.65) функционалы (2.60) и (2.62) имели следующий вид:

$$J_1 = \sum_{i=1}^4 \left(\int_0^{t_f} \left(x_1 + \frac{u^2}{2} \right) dt + p \right)_{\mathbf{x}^{0,i}} \rightarrow \min, \quad (2.66)$$

$$J_2 = \sum_{i=1}^4 \left(|x_i(t_f)| + p \right)_{\mathbf{x}^{0,i}} \rightarrow \min. \quad (2.67)$$

Задаем базисное решение в форме линейного пропорционального регулятора по отклонению каждой компоненты вектора состояний объекта от терминального состояния

$$h^0(x_1, x_2, x_3) = q_1(x_1^f - x_1) + q_2(x_2^f - x_2) + q_3(x_3^f - x_3),$$

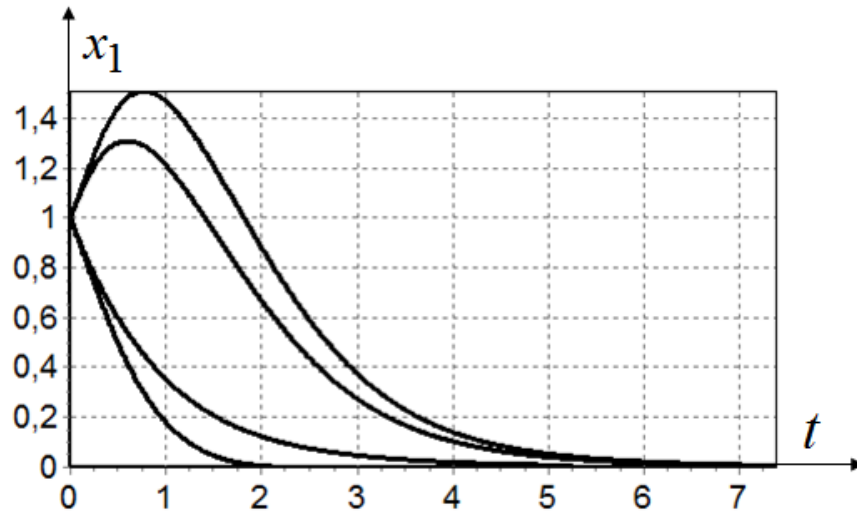
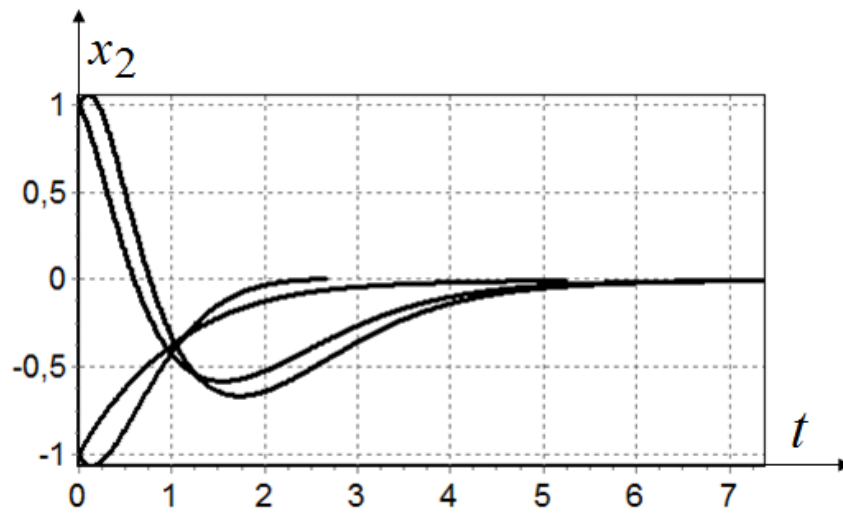
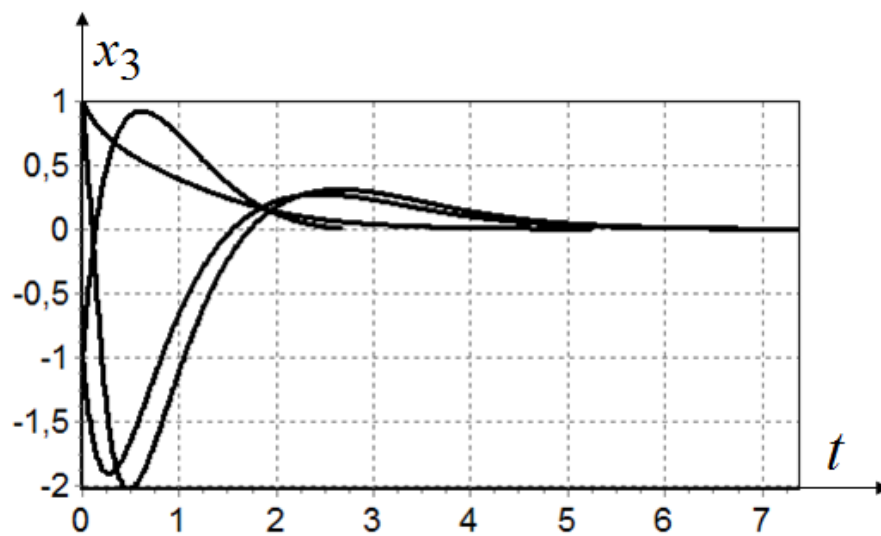
где $q_1 = 1$, $q_2 = 1$, $q_3 = 1$.

В результате применения метода вариационного аналитического программирования было получено решение в виде нелинейного регулятора, которое было отобрано из множества Парето как решение с нулевым значением функции штрафа (2.65) или как решение, обеспечивающее наилучшие значения функционалов без нарушения фазовых ограничений (2.56)

$$u = q_1(x_1^f - x_1) + q_2(x_2^f - x_2) + (x_3^f - x_3) \max \left\{ |x_1^f - x_1|, \left| q_1 - \sin(x_1^f - x_1) \right| \right\},$$

где $q_1 = 4,729600$, $q_2 = 8,815670$.

На рис. 2.1 - 2.4 приведены результаты моделирования системы с полученным решением для четырех начальных значений.

Рис. 2.1. График изменения переменной x_1 Рис. 2.2. График изменения переменной x_2 Рис. 2.3. График изменения переменной x_3

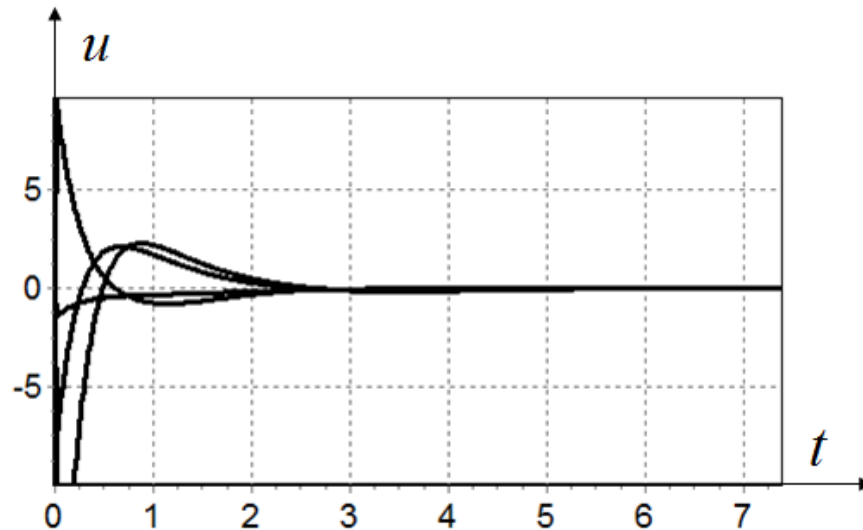


Рис. 2.4. График изменения управления u

Как видно из результатов моделирования полученная система управления обеспечивает достижение терминального состояния (2.55) всеми переменными из разных начальных состояний (2.61) без нарушения фазовых ограничений (2.56).

Характер изменения управления соответствует заданному критерию качества (2.60). По поведению управления видно, что оно стремится минимизировать площадь под кривыми или интеграл модуля функции управления.

2.4. Выводы к главе 2

1. Приведено описание метода вариационного аналитического программирования. Метод вариационного аналитического программирования использует форму кодирования математического выражения метода аналитического программирования и принцип малых вариаций базисного решения, заимствованный из метода сетевого оператора, для поиска оптимального решения. В результате метод вариационного аналитического программирования позволяет сократить пространство поиска за счет выбора базисного решения и использовать при поиске не только функции с одним или двумя аргументами, как в методе сетевого оператора.
2. Приведено описание малых вариаций кода аналитического программирования для математического выражения. Для корректного выполнения малых вариаций необходимо в записи кода находить коды математических подвыражений. Для определения кода подвыражений предложено использовать индекс кода, который вычисляет минимальное количество символов кода, которое должно находиться справа от текущего символа.
3. Решена задача синтеза системы управления для трехмерного линейного объекта с фазовыми ограничениями методом вариационного аналитического программирования. В результате решения задачи найдена нелинейная функция обратной связи, которая обеспечивает оптимальные траектории движения объекта в терминальное состояние из четырех начальных условий. Заданные фазовые ограничения при движении по оптимальным траекториям не нарушаются.

3. СИНТЕЗ СИСТЕМЫ УПРАВЛЕНИЯ ЛЕТАЮЩИМ РОБОТОМ

Рассмотрим задачу, в которой необходимо провести мониторинг области с помощью двух квадрокоптеров [19]. Пусть для области заданы координаты 23-х точек в трехмерном геометрическом пространстве. В рассматриваемой конкретной задаче точки имели следующие координаты:

$$P_0 = ([8,5 \ 18 \ 3,5]^T, [8,5 \ 15 \ 8,5]^T, [6 \ 11 \ 8,5]^T, [1 \ 12 \ 7]^T, [2 \ 10 \ 4,5]^T, \\ [5,5 \ 12 \ 5]^T, [5,5 \ 15 \ 3]^T, [0 \ 18 \ 0]^T, [8 \ 16 \ 4]^T, [16 \ 10 \ 4]^T, [20 \ 10 \ 10]^T, \\ [20 \ 10 \ 16]^T, [16 \ 12 \ 12]^T, [12 \ 16 \ 8]^T, [8 \ 18 \ 4]^T, [4 \ 20 \ 0]^T, [8 \ 24 \ -4]^T, \\ [16 \ 30 \ -8]^T, [20 \ 30 \ -10]^T, [16 \ 28 \ -12]^T, [12 \ 26 \ -8]^T, [8 \ 22 \ -4]^T, [4 \ 20 \ 0]^T).$$

В задаче специально с целью ее усложнения множество точек содержит как минимум две одинаковые точки. Задачу разбиваем на два этапа. Первоначально решаем задачу нахождения оптимальных траекторий, или в более точной формулировке задачу поиска путей движения двух квадрокоптеров в трехмерном пространстве.

В задаче все квадрокоптеры начинают движения из начала координат точки $[0 \ 0 \ 0]^T$. Каждому квадрокоптеру необходимо пройти половину точек и вернуться обратно в исходную начальную точку. Пусть первый квадрокоптер проходит первые 12 точек из 23-х, а второй квадрокоптер учитывает, что одна из точек повторяется и проходит остальные 11 точек.

Для решения задачи поиска оптимальных траекторий необходимо распределить заданные точки для каждого из квадрокоптеров. При случайном произвольном распределении точек в программе общая длина пути движения двух квадрокоптеров была равна величине 222,9 м. При этом первый квадрокоптер прошёл 114,3 м., а второй квадрокоптер прошёл 108,6 м.

Для оптимизации траекторий движения использовали вариационный генетический алгоритм. В результате работы алгоритма были получены траектории движения квадрокоптеров в форме упорядоченных множеств точек в трехмерном пространстве. Полученные множества имели следующий вид:

$$\begin{aligned} \tilde{P}_0 = & ([8,5 \ 18 \ 3,5]^T, [8 \ 18 \ 4]^T, [8 \ 16 \ 4]^T, [5,5 \ 15 \ 3]^T, [5,5 \ 12 \ 5]^T, \\ & [6 \ 11 \ 8,5]^T, [8,5 \ 15 \ 8,5]^T, [12 \ 16 \ 8]^T, [16 \ 12 \ 12]^T, [20 \ 10 \ 16]^T, [20 \ 10 \ 10]^T, \\ & [16 \ 10 \ 4]^T, [2 \ 10 \ 4,5]^T, [1 \ 12 \ 7]^T, [0 \ 18 \ 0]^T, [4 \ 20 \ 0]^T, [4 \ 20 \ 0]^T, \\ & [8 \ 22 \ -4]^T, [8 \ 24 \ -4]^T, [12 \ 26 \ -8]^T, [16 \ 30 \ -8]^T, [20 \ 30 \ -10]^T, [16 \ 28 \ -12]^T). \end{aligned}$$

Согласно предположению об оптимальном распределении точек генетическим алгоритмом две одинаковые точки оказались расположены рядом, поскольку между ними было нулевое расстояние.

В результате общая длина оптимальных траекторий движения квадрокоптеров составила 178,4 м. Из множества точек упорядоченных оптимально по общей длине пути первые 11 первых точек относились к пути первого квадрокоптера, длина пути при этом для первого квадрокоптера составила 86,7 м. Длина остальных точек для второго квадрокоптера составила 91,7 м.

Для устойчивого состояния квадрокоптера в установленной в пространстве состояний в любой точке на втором этапе решали задачу синтеза системы управления. Для решения задачи использовали разработанный в диссертации метод вариационного аналитического программирования. Движение квадрокоптера по траектории осуществляется за счет переключения точек устойчивого состояния в системе его управления.

3.1 Задача синтеза оптимального управления летающим роботом

Математическая модель квадрокоптера в случае отсутствия вращения вокруг вертикальной оси описывается следующей системой дифференциальных уравнений [75]

$$\ddot{x} = -\frac{T}{m} \sin \theta, \quad (3.1)$$

$$\ddot{y} = \frac{T}{m} \cos \gamma \cos \theta - g, \quad (3.2)$$

$$\ddot{z} = \frac{T}{m} \cos\theta \sin\gamma, \quad (3.3)$$

$$\ddot{\gamma} = \frac{M_x}{I_x}, \quad (3.4)$$

$$\ddot{\theta} = \frac{M_z}{I_z}, \quad (3.5)$$

где

$$T = |u_1| + |u_2| + |u_3| + |u_4|, \quad (3.6)$$

$$M_x = (u_1 + u_2 - u_3 - u_4)l, \quad (3.7)$$

$$M_z = (u_1 - u_2 - u_3 + u_4)l, \quad (3.8)$$

u_i - сила тяги пропеллера i , $i = 1, 2, 3, 4$, $g = 9,81$ м/с², I_x - момент инерции по оси ОХ, $I_x = 0,03$ кгм², I_z - момент инерции по оси ОZ, $I_z = 0,03$ кгм², m - масса квадрокоптера, $m = 1$ кг, l - расстояние между противоположными пропеллерами, $l = 1$ м, M_x - момент вокруг оси ОХ, M_z - момент вокруг оси ОZ, T - сила тяги всех пропеллеров квадрокоптера.

Целью управления являются координаты точек в пространстве состояний. Траектория движения квадрокоптера представляет собой упорядоченное множество векторов из значений трех компонент положения точек

$$P = \left([x_1 \ y_1 \ z_1]^T, \dots, [x_M \ y_M \ z_M]^T \right). \quad (3.9)$$

В качестве критерия управления используем функционалы

$$J_1 = \sum_{i=1}^M \sqrt{(x(t_i) - x_i)^2 + (y(t_i) - y_i)^2 + (z(t_i) - z_i)^2} \rightarrow \min, \quad (3.10)$$

$$J_2 = t_M \rightarrow \min, \quad (3.11)$$

где

$$t_i = \begin{cases} t, & \text{если } \left(\sqrt{(x(t) - x_i)^2 + (y(t) - y_i)^2 + (z(t) - z_i)^2} < \delta \right) \wedge (t - t_{i-1} < \Delta t), \\ t_{i-1} + \Delta t, & \text{если } \Delta t \leq t - t_{i-1} \end{cases},$$

$$i \leftarrow i + 1, \text{ если } (i < M) \wedge \left(\sqrt{(x(t) - x_i)^2 + (y(t) - y_i)^2 + (z(t) - z_i)^2} < \delta \right) \vee (\Delta t \leq t - t_{i-1}),$$

$\Delta t = 8 \text{ с.}, \delta = 0,25 \text{ м.}, t_0 = 0.$

В прикладной задаче для системы установлены следующие начальные условия: $x(0) = \pm 1 \text{ м.}, \dot{x}(0) = 0 \text{ м/с.}, y(0) = 20 \pm 1 \text{ м.}, \dot{y}(0) = 0 \text{ м/с.}, z(0) = 0 \text{ м.}, \dot{z}(0) = 0 \text{ м/с.}, \gamma(0) = 0 \text{ рад.}, \dot{\gamma}(0) = 0 \text{ рад/с.}, \theta(0) = 0 \text{ рад.}, \dot{\theta}(0) = 0 \text{ рад/с.}$

Для значений управления заданы ограничения

$$T^- \leq T \leq T^+, \quad (3.12)$$

$$M_x^- \leq M_x \leq M_x^+, \quad (3.13)$$

$$M_z^- \leq M_z \leq M_z^+, \quad (3.14)$$

где $T^- = 6 \text{ Н}, T^+ = 16 \text{ Н}, M_x^- = -0,1 \text{ Нм}, M_x^+ = 0,1 \text{ Нм}, M_z^- = -0,1 \text{ Нм}, M_z^+ = 0,1 \text{ Нм},$

При решении задачи синтеза систем управления использовалось объединённое множество, включающее переменные, параметры и элементарные функции следующего вида [19]:

$$F = \left(f_1 = x_i^f - x, f_2 = y_i^f - y, f_3 = z_i^f - z, f_4 = \dot{x}, f_5 = \dot{y}, f_6 = \dot{z}, f_7 = \gamma, f_8 = \dot{\gamma}, \right. \\ \left. f_9 = \theta, f_{10} = \dot{\theta}, f_{11} = q_1, f_{12} = q_2, f_{13} = q_3, f_{14} = q_4, f_{15} = q_5, f_{16} = q_6, f_{17} = q_7, \right. \\ \left. f_{18} = q_8, f_{19} = q_9, f_{20} = q_{10}, f_{21} = q_{11}, \right.$$

$$f_{22}(a) = \text{sgn}(a)a^2, f_{23}(a) = a^2, f_{24}(a) = -a,$$

$$f_{25}(a) = \text{sgn}(a)\sqrt{|a|}, f_{26}(a) = a^{-1}, f_{27}(a) = e^a, f_{28}(a) = \ln|a|, f_{29}(a) = \frac{1 - e^{-a}}{1 + e^{-a}},$$

$$f_{30}(a) = \begin{cases} 1, & \text{если } a \geq 0 \\ 0, & \text{иначе} \end{cases}, f_{31}(a) = \text{sgn}(a), f_{32}(a) = \cos(a), f_{33}(a) = \sin(a),$$

$$f_{34}(a) = \arctan(a), f_{35}(a) = a^3, f_{36}(a) = \sqrt[3]{a}, f_{37}(a) = \begin{cases} a, & \text{если } |a| \leq 1 \\ \text{sgn}(a), & \text{иначе} \end{cases},$$

$$f_{38}(a) = \text{sgn}(a)\ln(|a| + 1), f_{39}(a) = \text{sgn}(a)\left(e^{-|a|} - 1\right), f_{40}(a) = \text{sgn}(a)\left(1 - e^{-|a|}\right),$$

$$f_{41}(a) = a/2, f_{42}(a) = 2a, f_{43}(a) = e^{|a|}, f_{44}(a) = a - a^3,$$

$$\begin{aligned}
f_{45}(a,b) &= a + b, f_{46}(a,b) = ab, f_{47}(a,b) = \max\{a,b\}, f_{48}(a,b) = \min\{a,b\}, \\
f_{49}(a,b) &= a + b - ab, f_{50}(a,b) = \operatorname{sgn}(a+b)\sqrt{a^2 + b^2}, f_{51}(a,b) = \operatorname{sgn}(a+b)(|a| + |b|), \\
f_{52}(a,b) &= \operatorname{sgn}(a+b)\max\{|a|, |b|\}, \\
f_{53}(a,b,c) &= \begin{cases} b, & \text{если } a \leq 0 \\ c, & \text{иначе} \end{cases}, f_{54}(a,b,c) = \begin{cases} c, & \text{если } a \leq b \\ -c, & \text{иначе} \end{cases}, \\
f_{55}(a,b,c) &= \begin{cases} b, & \text{если } a < b \\ c, & \text{если } (a \geq b) \wedge (a > c), \\ a, & \text{если } (a \geq b) \wedge (a \leq c) \end{cases}, \\
f_{56}(a,b,c) &= \begin{cases} b - a, & \text{если } |a - b| < |a - c| \\ c - a, & \text{иначе} \end{cases}, \tag{3.15}
\end{aligned}$$

Для решения задачи синтеза методом вариационного аналитического программирования необходимо знать количество функций с определенным числом аргументов в объединенном множестве. Для рассматриваемого примера объединенное множество (3.15) включало $m_0 = 21$ функций без аргументов или параметров и координат вектора состояний, функций с одним аргументом было $m_1 = 23$, функций с двумя аргументами $m_2 = 8$ и функций с тремя аргументами $m_3 = 4$.

3.2 Вычислительный эксперимент

Для решения задачи использовался генетический алгоритм со следующими параметрами: длина кода одной компоненты управления 32 позиции, количество возможных решений в начальном множестве решений 1024, количество поколений 128, количество предполагаемых скрещиваний в одном поколении 64, число функционалов качества 2, количество малых вариаций в одном наборе для описания одного возможного решения 8, число бит под целую часть значения параметра 4, число бит под дробную часть параметра 12, число поколений между сменой эпохи или сменой базисного решения 24, количество элитарных возможных решения 8, параметр, используемый для вычисления вероятности скрещивания 0,4, величина вероятности мутации 0,7.

В эксперименте рассмотрено движение квадрокоптера по двум заданным пространственным траекториям. В результате применения вариационного аналитического программирования и выбора из множества Парето получено следующее оптимальное управление:

$$T = \begin{cases} T^+, & \text{если } \tilde{T} > T^+ \\ T^-, & \text{если } \tilde{T} < T^-, \\ \tilde{T}, & \text{иначе} \end{cases} \quad (3.16)$$

$$M_z = \begin{cases} M_z^+, & \text{если } \tilde{M}_z > M_z^+ \\ M_z^-, & \text{если } \tilde{M}_z < M_z^-, \\ \tilde{M}_z, & \text{иначе} \end{cases} \quad (3.17)$$

$$M_x = \begin{cases} M_x^+, & \text{если } \tilde{M}_x > M_x^+ \\ M_x^-, & \text{если } \tilde{M}_x < M_x^-, \\ \tilde{M}_x, & \text{иначе} \end{cases} \quad (3.18)$$

где

$$\tilde{T} = \Delta_x + \sqrt[3]{\Delta_y} - \Delta_x \sqrt[3]{\Delta_y}, \quad (3.19)$$

$$\tilde{M}_z = \operatorname{sgn}(\Delta_y) e^{-|\Delta_y|} \operatorname{sgn}(\mu(2\Delta_x)) \left(\ln(|\mu(2\Delta_x)| + 1) - \ln^3(|\mu(2\Delta_x)| + 1) \right) \quad (3.20)$$

$$\tilde{M}_x = \frac{1 - e^{-A}}{1 + e^{-A}}, \quad (3.21)$$

$$\mu(a) = \begin{cases} a, & \text{если } |a| < 1 \\ \operatorname{sgn}(a) - \text{иначе} \end{cases}$$

$$A = \min \left\{ \operatorname{sgn}(\Delta_z) \ln(|\Delta_z| + 1), \min \left\{ \max \left\{ \operatorname{sgn}(B) \ln(|B|), \Delta_x \right\}, q_{11} \right\} \right\},$$

$$B = f_{3,1}(\ln(q_{12} + 1), \Delta_z, \Delta_z) \ln \left(\left| f_{3,1}(\ln(q_{12} + 1), \Delta_z, \Delta_z) \right| + 1 \right),$$

$$f_{3,1}(a, b, c) = \begin{cases} c, & \text{если } a < b \\ -c & \text{иначе} \end{cases},$$

$$\Delta_y = -q_{10} \dot{y} - q_9 (y - y_i) + g + \operatorname{arctg}(\dot{y}) + \sin(\dot{y}) + (q_6 \theta)^2 - \operatorname{sgn}(y - y_i) \sqrt{q_9 |y - y_i|},$$

$$\Delta_x = -q_5 \dot{\theta} - q_6 \theta + q_7 \dot{x} + q_8 (x - x_i),$$

$$\Delta_z = -q_1 \dot{\gamma} - q_2 \gamma - q_3 \dot{z} - q_4 (z - z_i),$$

$$q_1 = 1,075, q_2 = 3, q_3 = 0,527, q_4 = 0,324, q_5 = 1,075, q_6 = 3, q_7 = 0,527, \\ q_8 = 0,324, q_9 = 15,51, q_{10} = 15,19, q_{11} = 3,234, q_{12} = 1,5625.$$

В первом вычислительном эксперименте использовалось $M = 12$ точек пространственной траектории

$$\tilde{P}_0 = ([8,5 \ 18 \ 3,5]^T, [8 \ 18 \ 4]^T, [8 \ 16 \ 4]^T, [5,5 \ 15 \ 3]^T, [5,5 \ 12 \ 5]^T, \\ [6 \ 11 \ 8,5]^T, [8,5 \ 15 \ 8,5]^T, [12 \ 16 \ 8]^T, [16 \ 12 \ 12]^T, [20 \ 10 \ 16]^T, \\ [20 \ 10 \ 10]^T, [16 \ 10 \ 4]^T).$$

Результаты моделирования для четырех начальных условий

- 1) $x(0) = +1\text{ м}, y(0) = 21\text{ м};$
- 2) $x(0) = -1\text{ м}, y(0) = 21\text{ м};$
- 3) $x(0) = +1\text{ м}, y(0) = 19\text{ м};$
- 4) $x(0) = -1\text{ м}, y(0) = 19\text{ м};$

представлены на рис. 3.1 – 3.9.

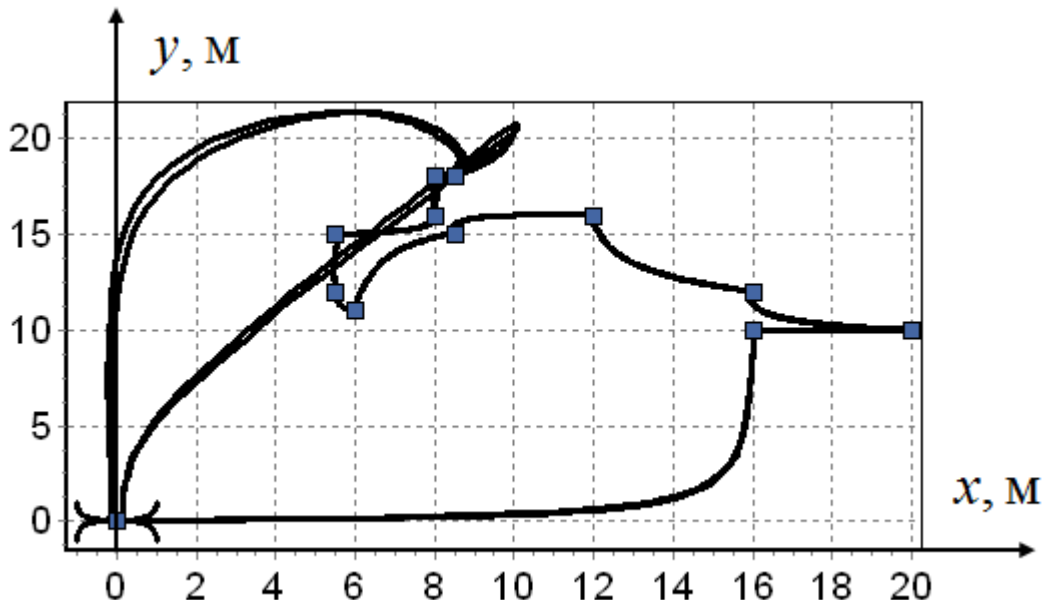


Рис. 3.1. Траектории движения в вертикальной плоскости $\{y, x\}$

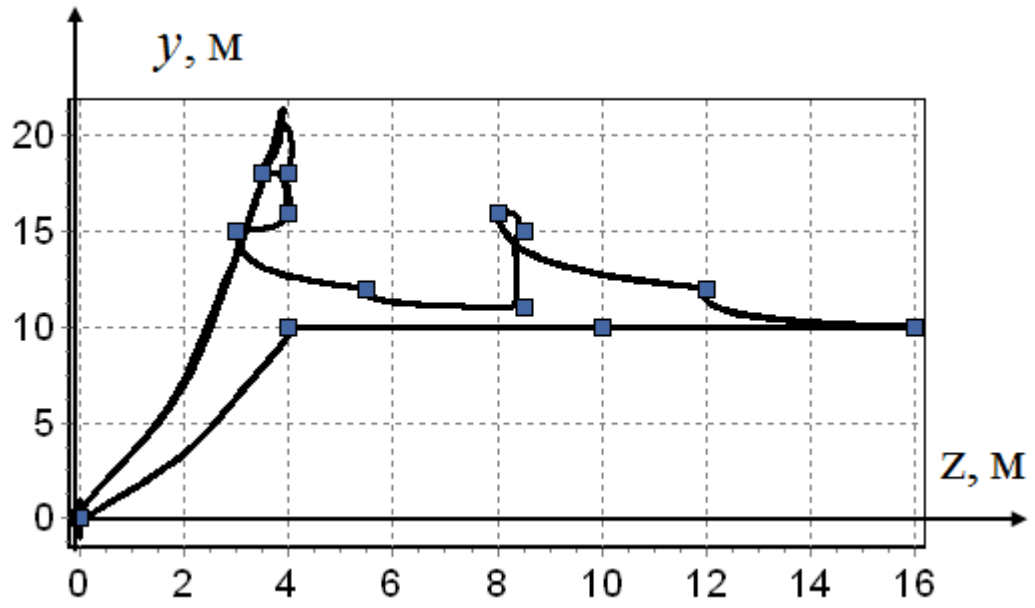


Рис. 3.2. Траектории движения в вертикальной плоскости $\{y, z\}$

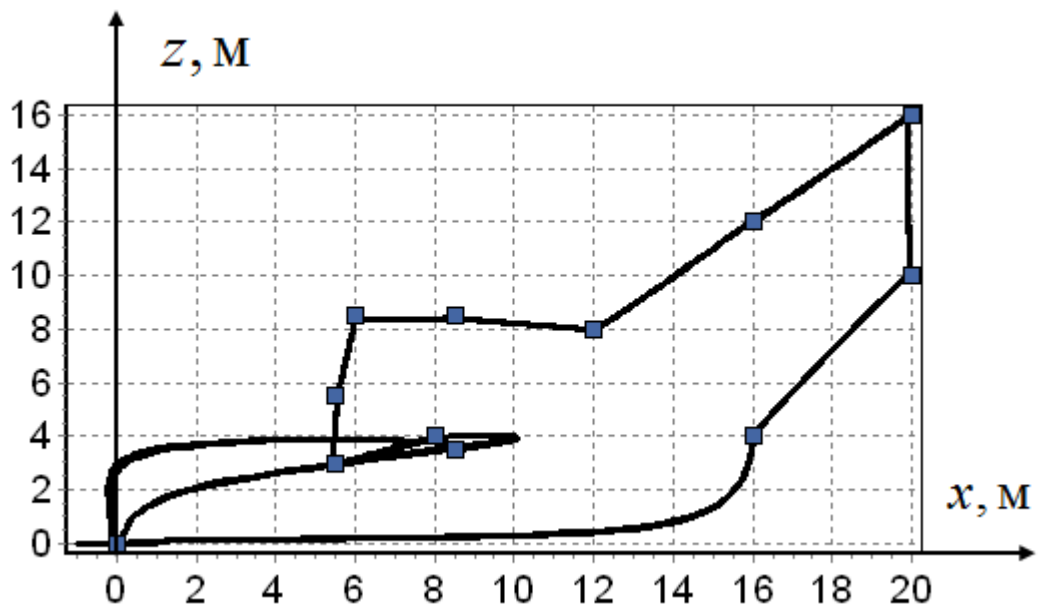
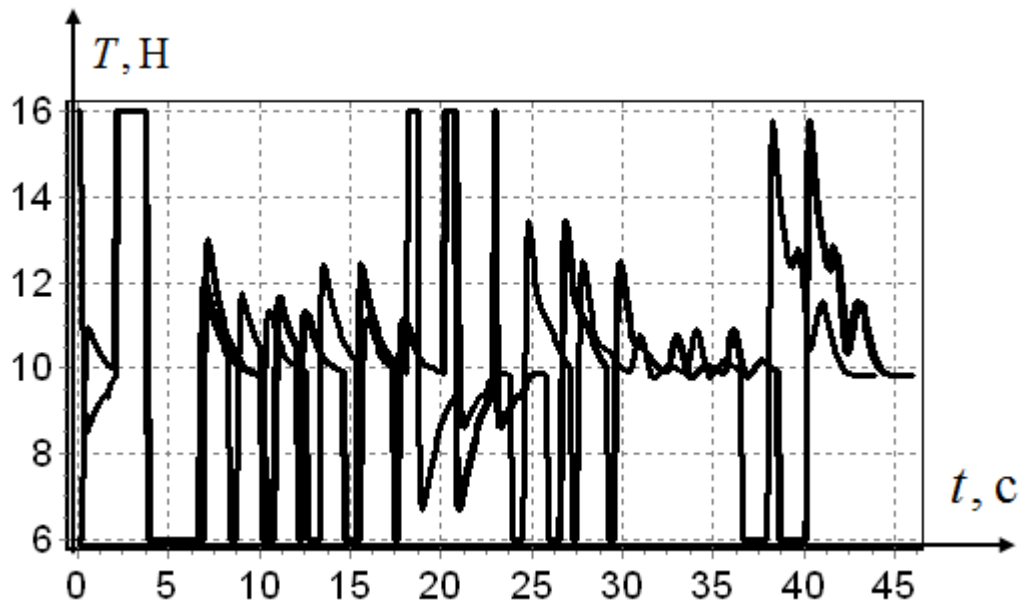
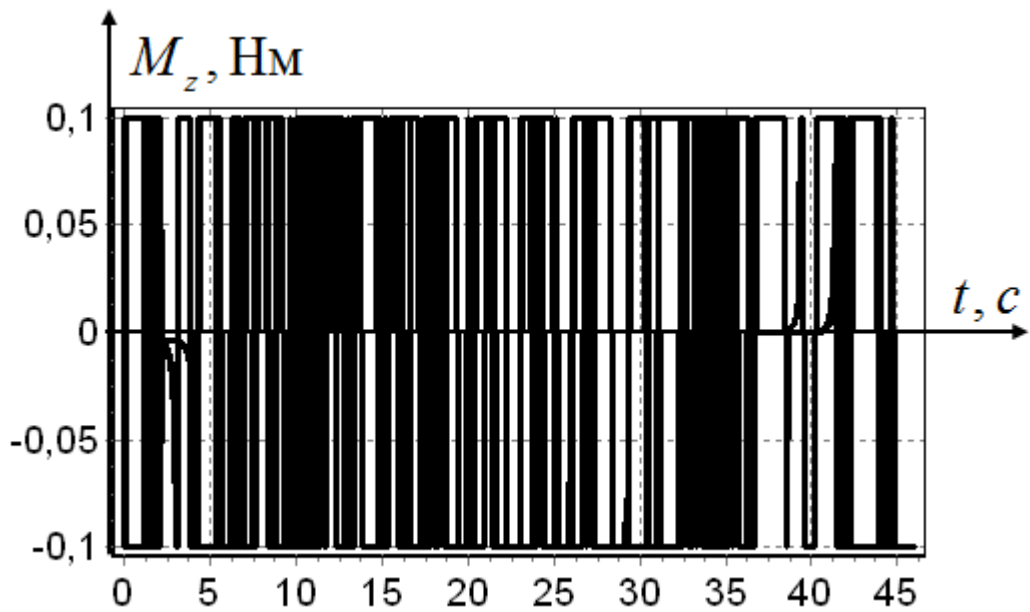


Рис. 3.3. Траектории движения в горизонтальной плоскости $\{z, x\}$

Рис. 3.4. Управляющая суммарная тяга T Рис. 3.5. Управляющий момент M_z

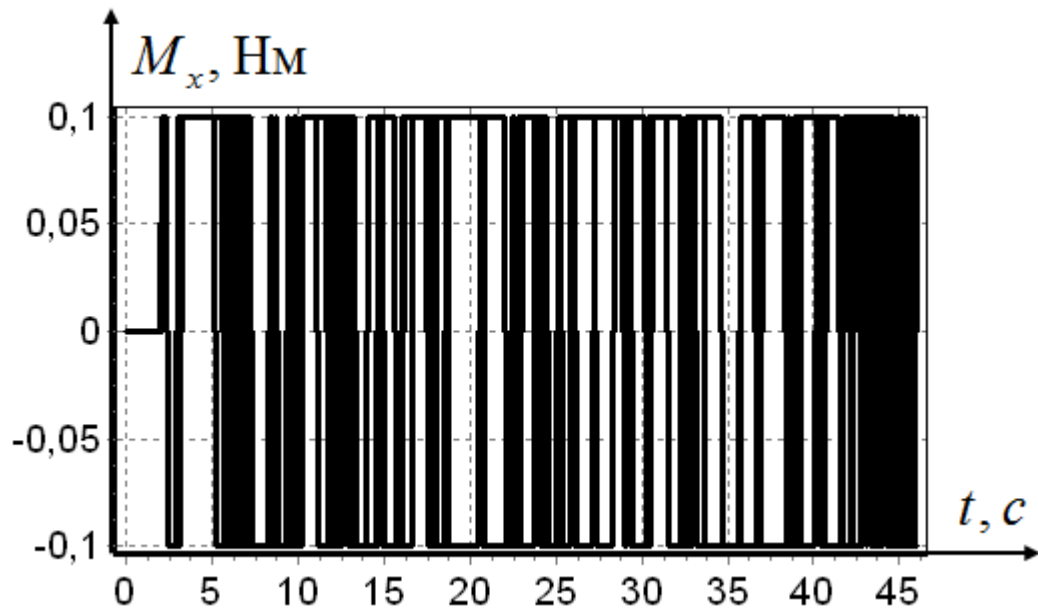


Рис. 3.6. Управляющий момент M_x

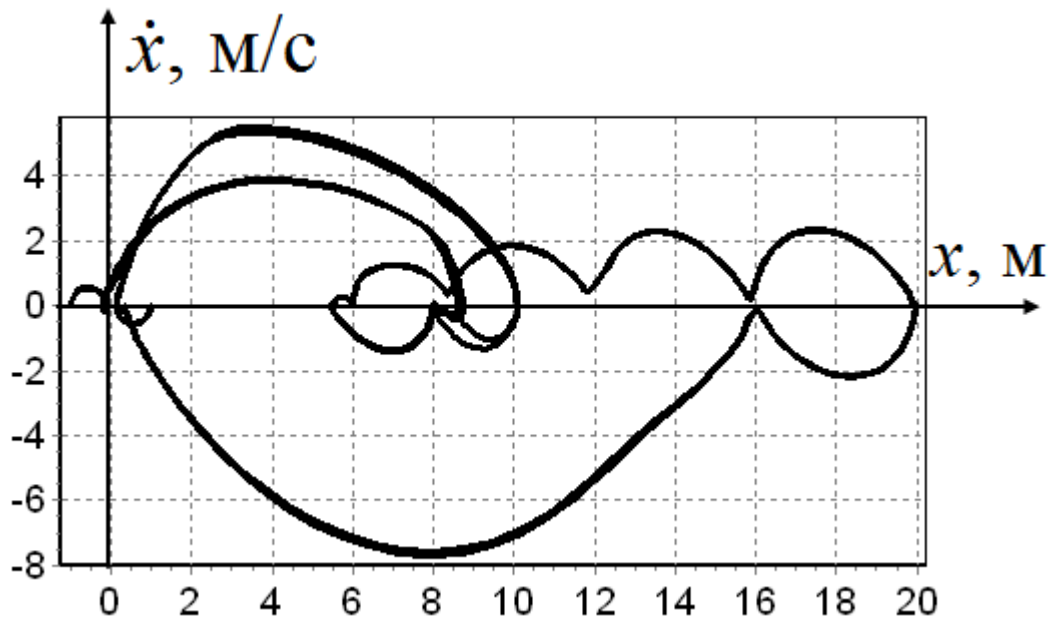
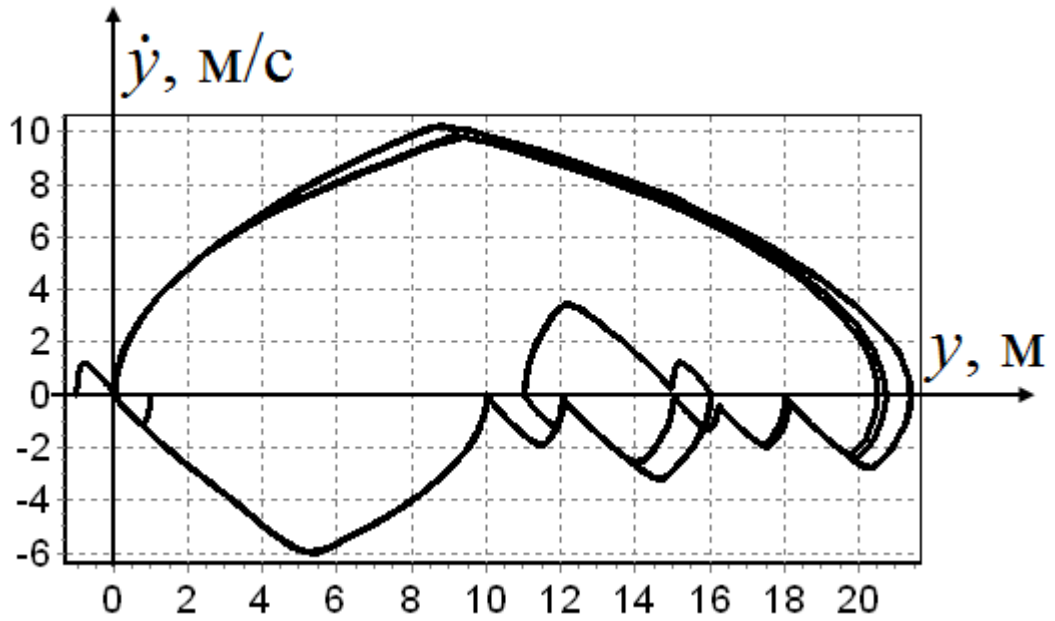
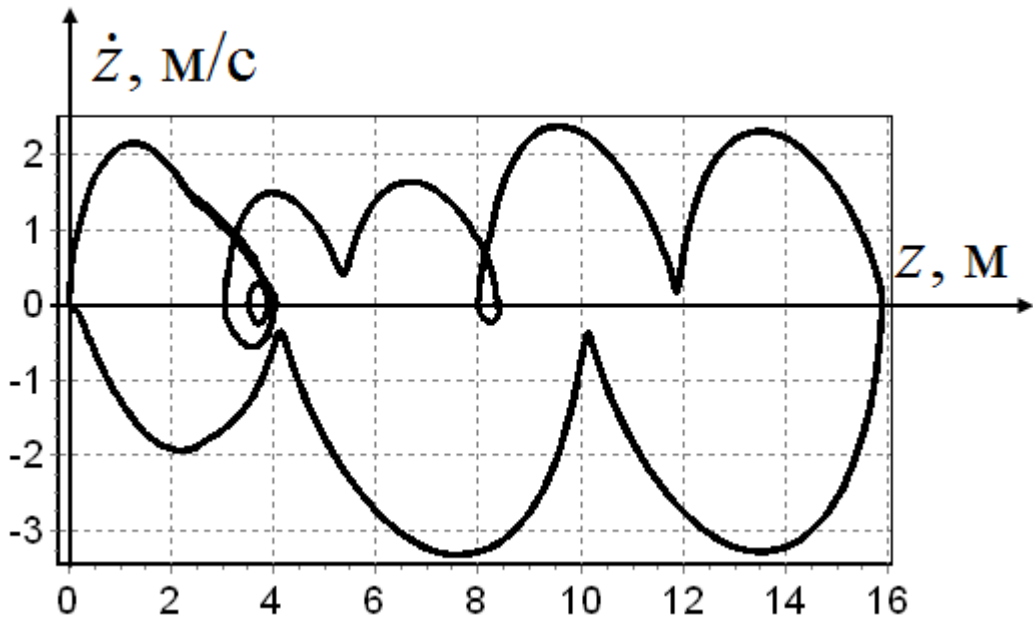


Рис. 3.7. Фазовый портрет координаты x

Рис. 3.8. Фазовый портрет координаты y Рис. 3.9. Фазовый портрет координаты z

Как видим из рисунков начальные значения влияют только на точность достижения первой точки. Характер управления для всех начальных условий приблизительно одинаковый. Точность движения по траектории достаточно высока и не зависит от начальных условий.

Во втором вычислительном эксперименте рассматривалось движение второго квадрокоптера по остальным $M = 11$ точкам траектории

$$\tilde{P}_0 = ([2 \ 10 \ 4,5]^T, [1 \ 12 \ 7]^T, [0 \ 18 \ 0]^T, [4 \ 20 \ 0]^T, [4 \ 20 \ 0]^T, \\ [8 \ 22 \ -4]^T, [8 \ 24 \ -4]^T, [12 \ 26 \ -8]^T, [16 \ 30 \ -8]^T, [20 \ 30 \ -10]^T, [16 \ 28 \ -12]^T).$$

Результаты моделирования синтезированной системы управления для тех же начальных условий приведены на рис. 3.10-3.29.

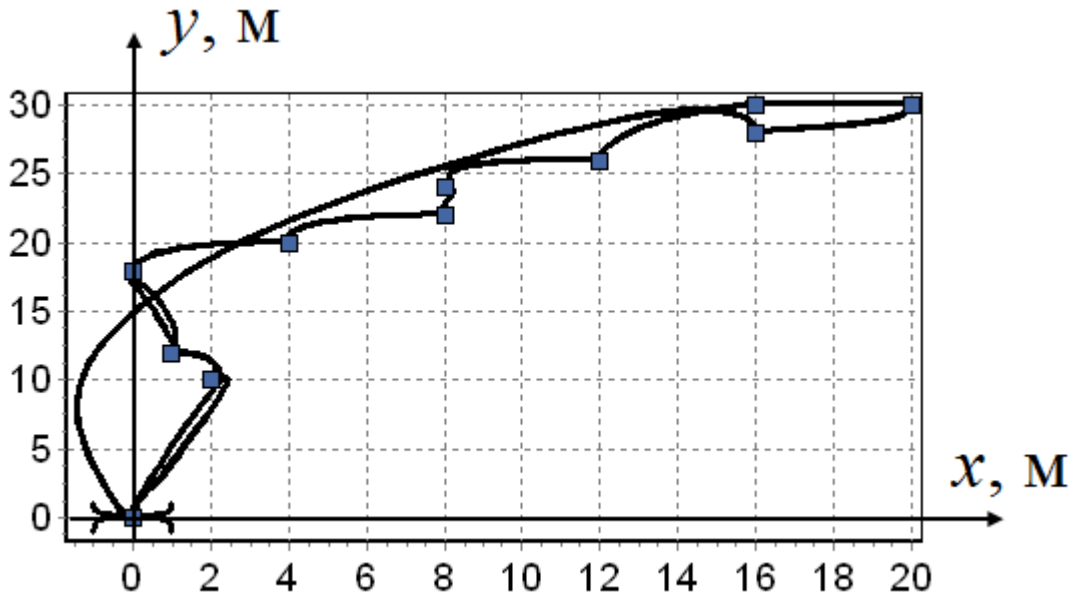


Рис. 3.10. Движение летающего робота в вертикальной плоскости $\{x, y\}$

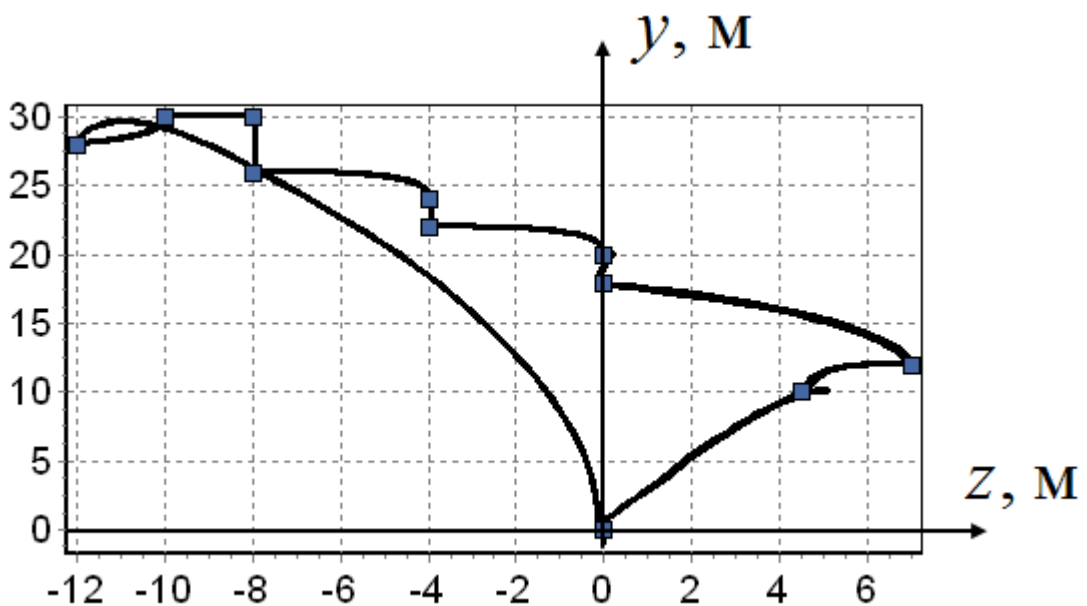


Рис. 3.11. Движение летающего робота в вертикальной плоскости $\{y, z\}$

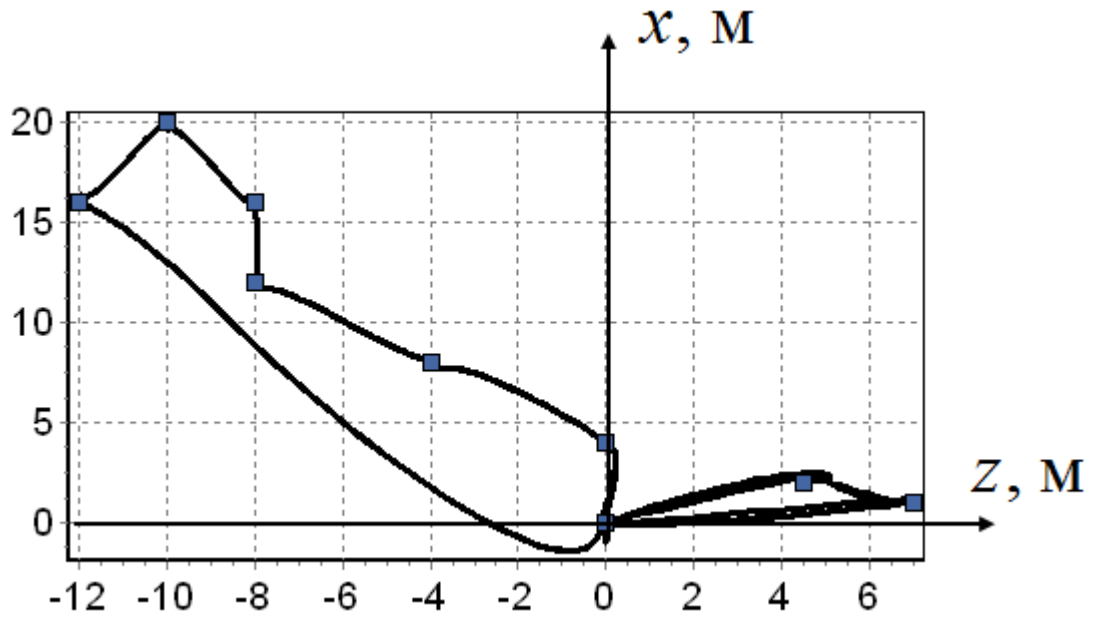


Рис. 3.12. Движение летающего робота в горизонтальной плоскости $\{x, z\}$

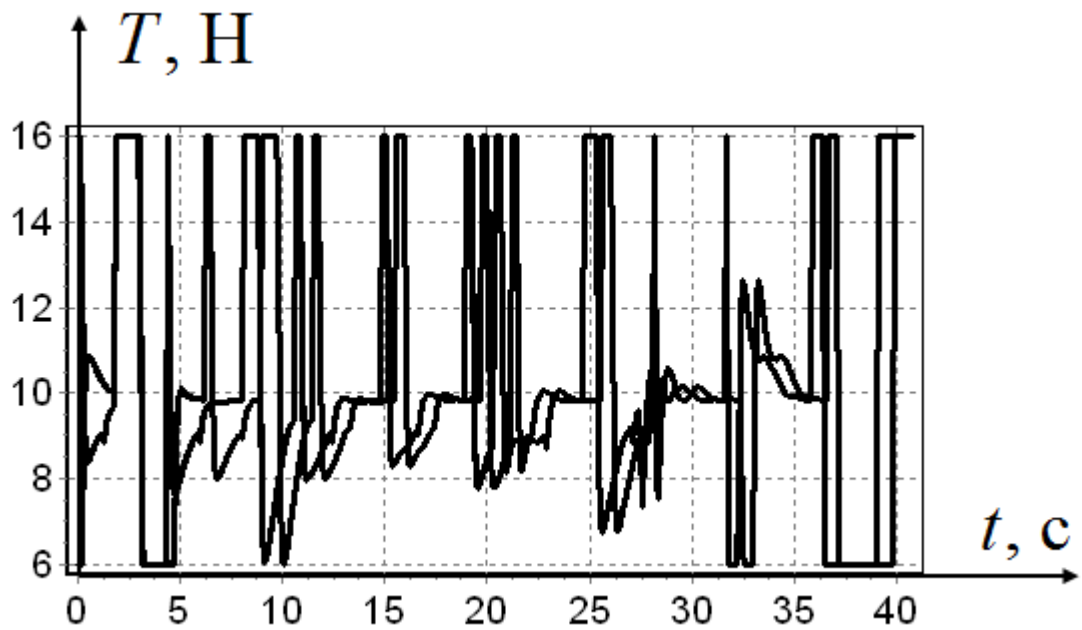
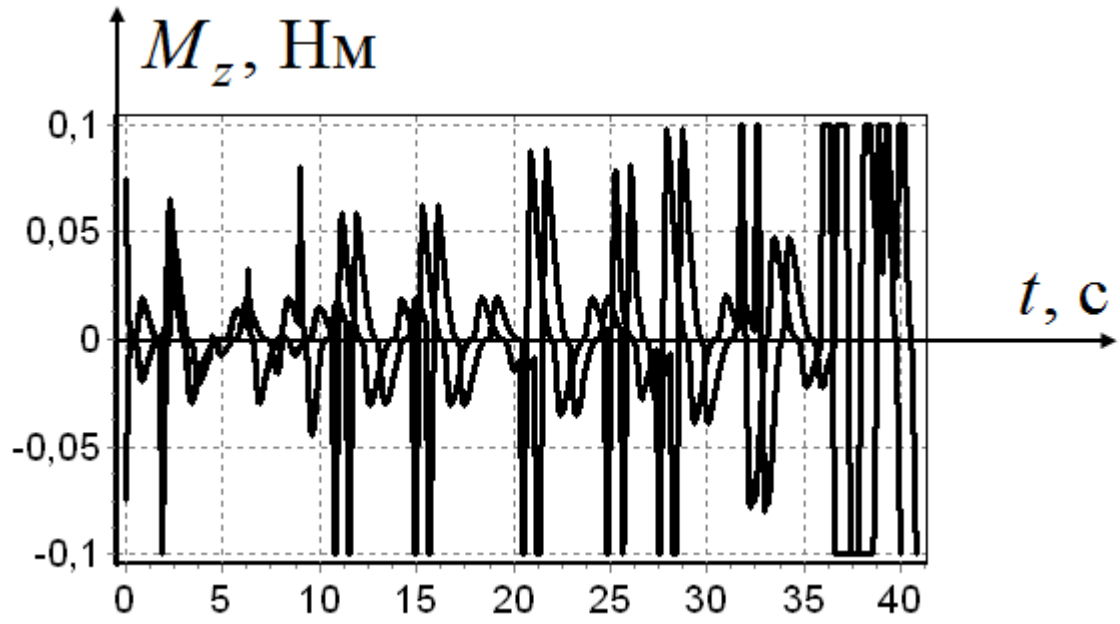
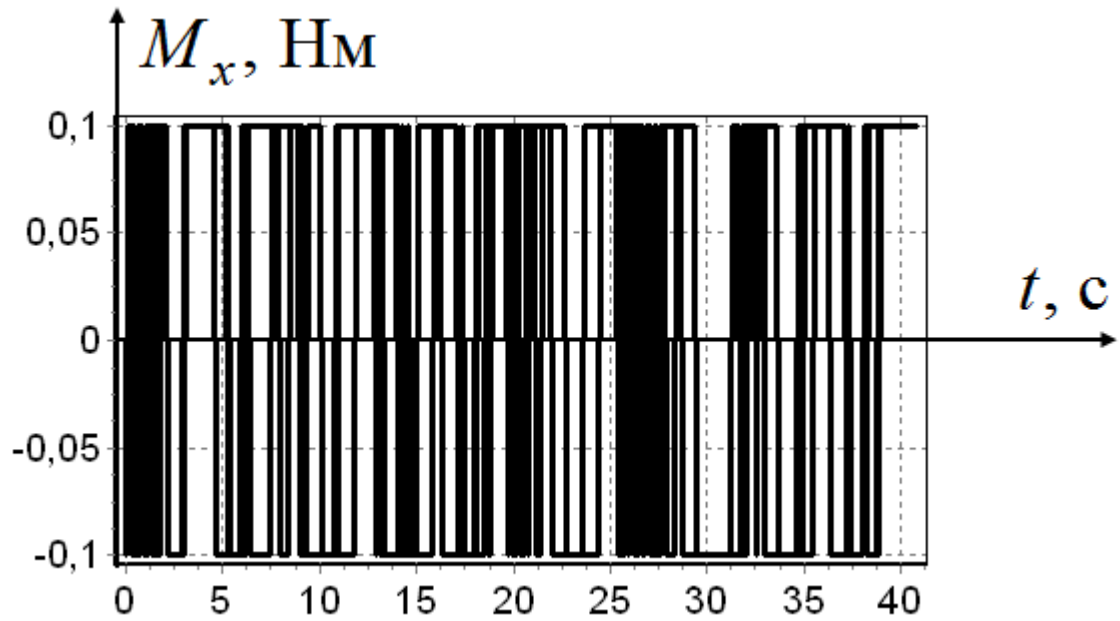
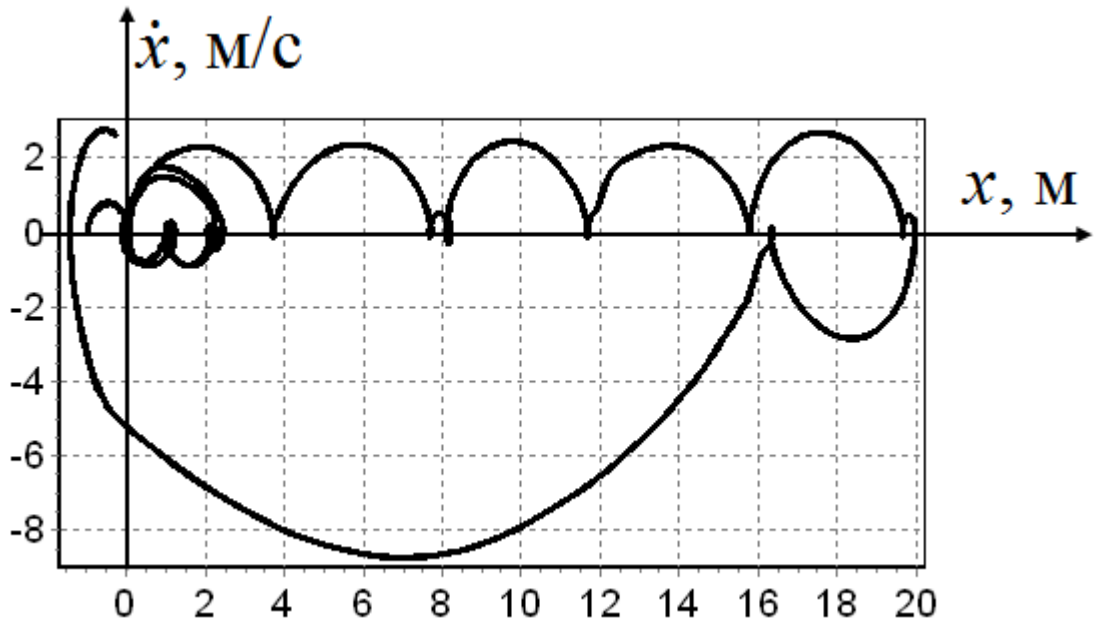
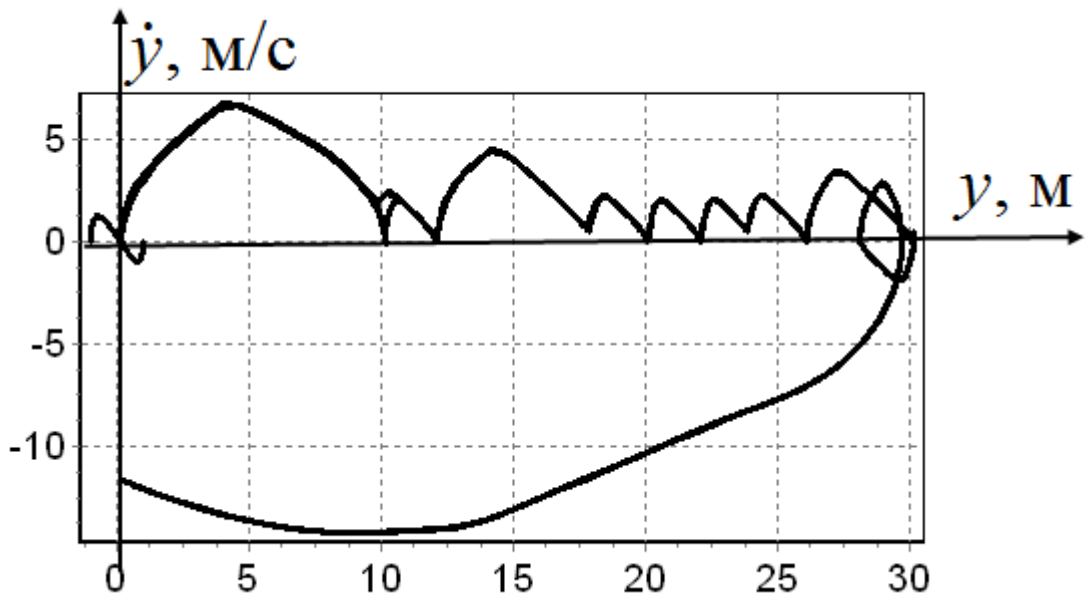


Рис. 3.13. Управляющая суммарная тяга T

Рис. 3.14. Управляющий момент M_z Рис. 3.15. Управляющий момент M_x

Рис. 3.16. Фазовый портрет координаты x Рис. 3.17. Фазовый портрет координаты y

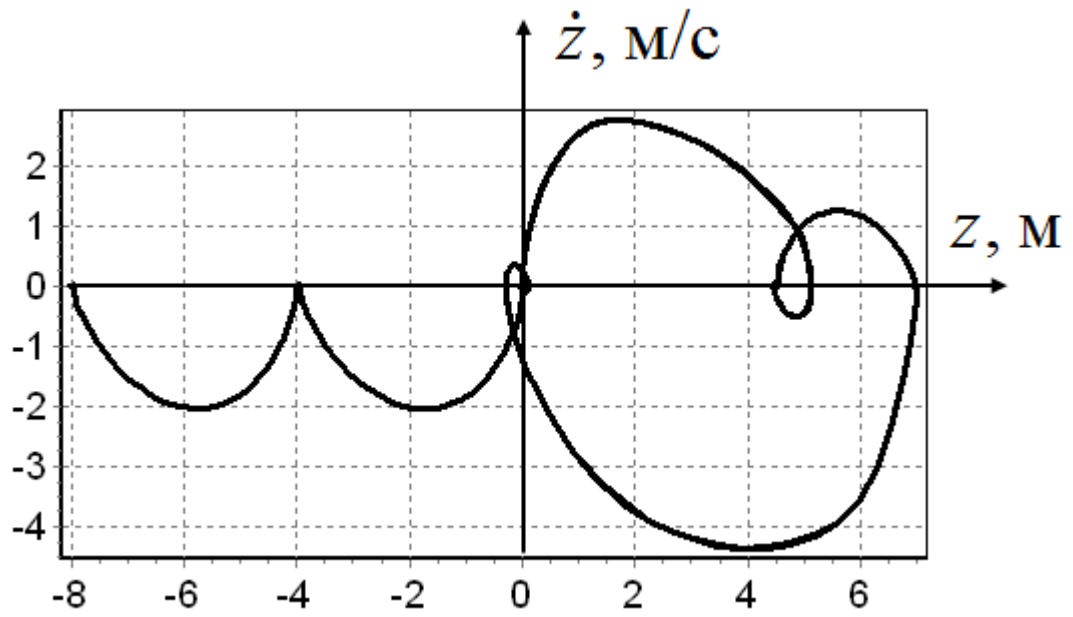
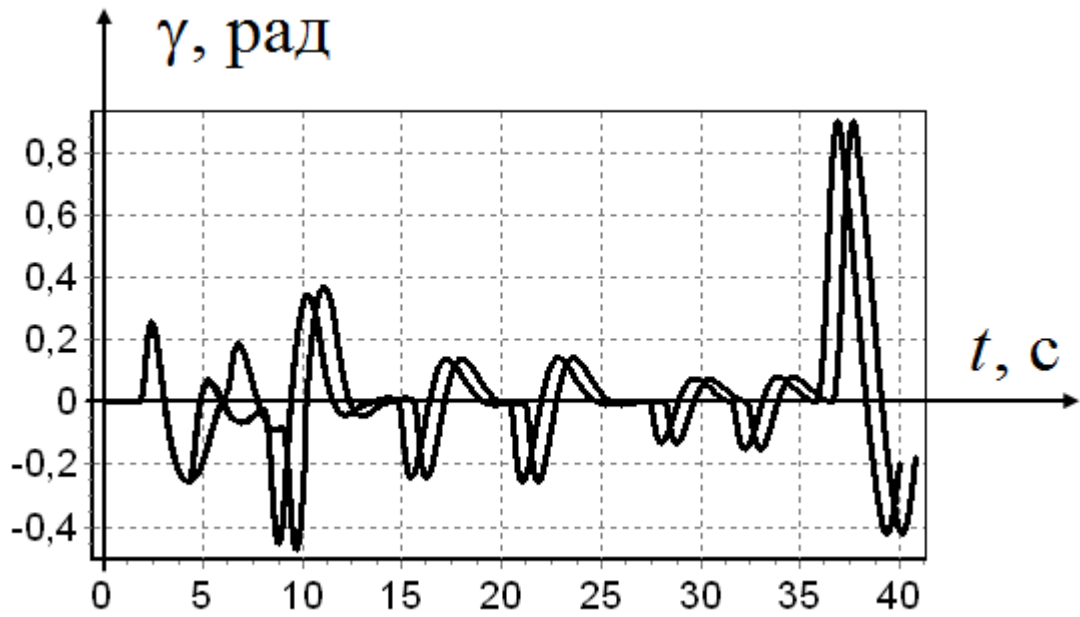
Рис. 3.18. Фазовый портрет координаты z 

Рис. 3.19. Угол крена

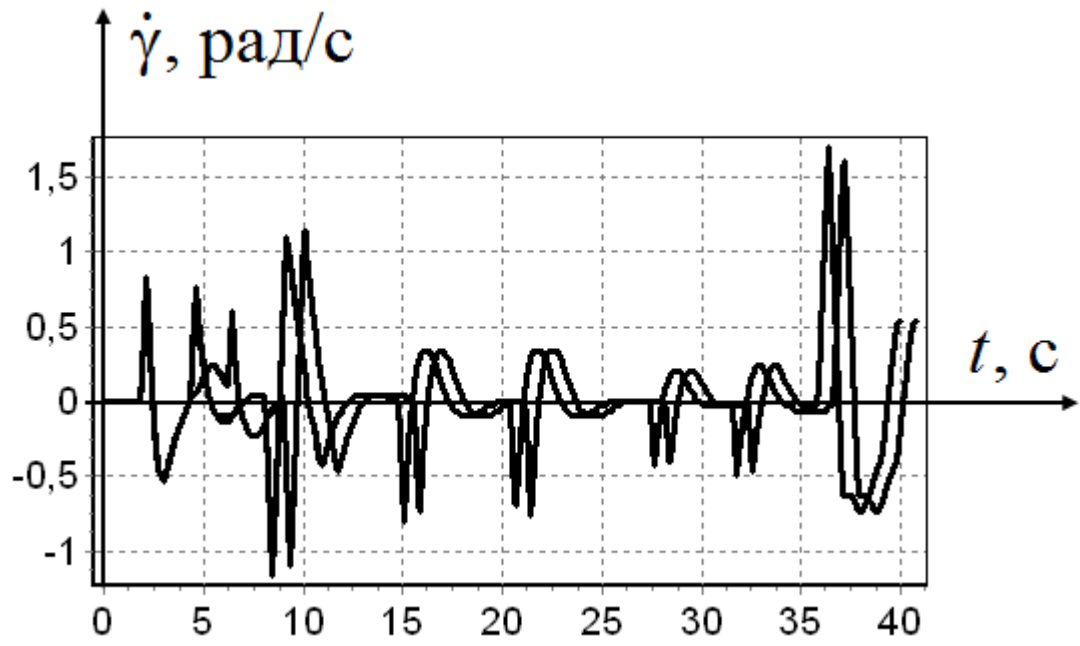


Рис. 3.20. Скорость изменения угла крена

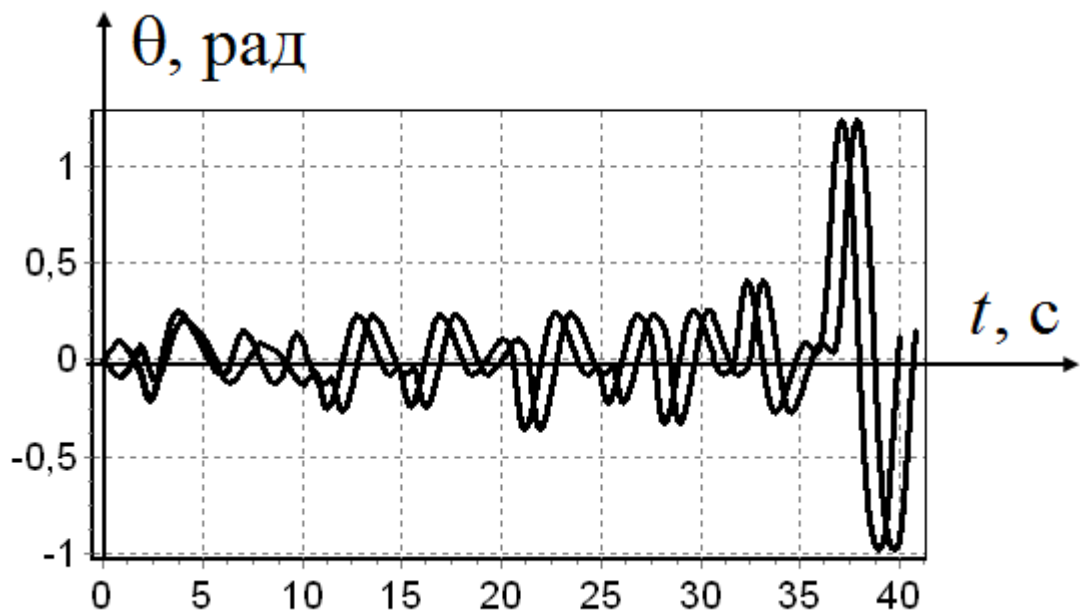


Рис. 3.21. Угол тангажа

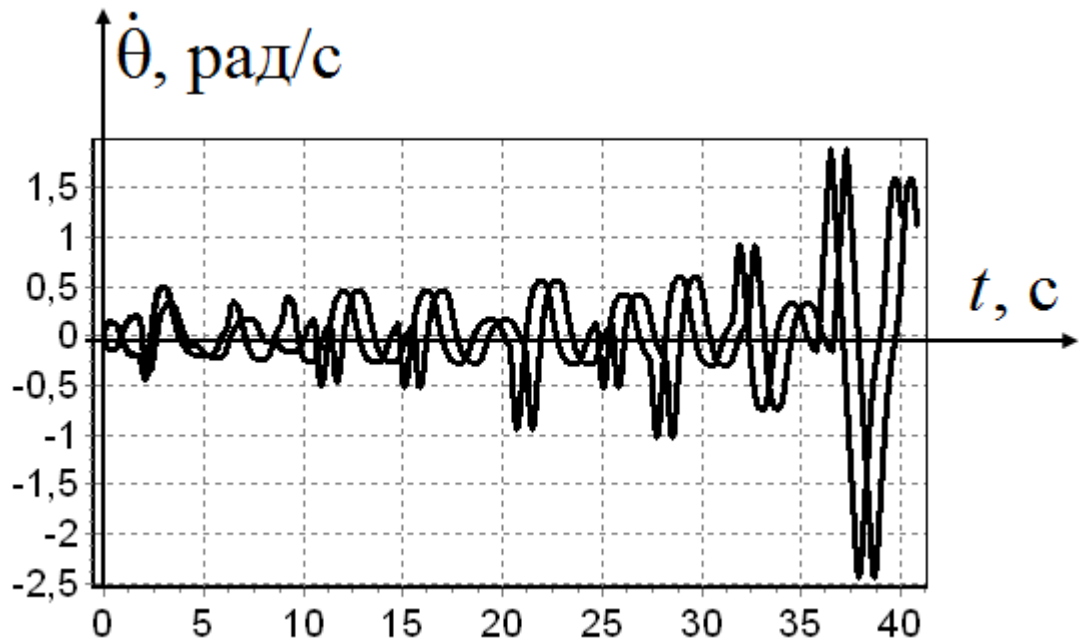


Рис. 3.22. Скорость угла тангажа

Результаты моделирования показали, что полученная с помощью автоматического синтеза система управления обеспечивает качественное движение объекта по пространственной траектории. Синтезированная методом вариационного аналитического программирования система управления пригодна для использования в практических системах и не уступает по качеству управления системам, полученным вручную на основе технического синтеза специалистами и разработчиками систем автоматического управления.

3.3 Выводы к главе 3

1. С помощью вариационного генетического алгоритма решена задача коммивояжера для поиска оптимальных траекторий для двух квадрокоптеров, в результате заданные точки в трехмерном пространстве были упорядочены так, что суммарная длина путей для двух квадрокоптеров сократилась более, чем на 20 % по сравнению с путем, определяемым первоначальным порядком точек в трехмерном пространстве.
2. Решена задача синтеза системы управления летающим роботом методом вариационного аналитического программирования. В результате решения задачи получена нелинейная обратная связь по всему вектору состояния объекта управления.
3. С помощью моделирования движения объекта управления по двум пространственным траекториям из двенадцати и одиннадцати точек показано, что полученная при синтезе система управления обеспечивает точное движение объекта управления по пространственной траектории. Точность движения не зависит от заданных начальных условий.

ЗАКЛЮЧЕНИЕ

1. Разработана методика решения задачи мониторинга местности группой летающих роботов, квадрокоптеров. Методика включает решение задачи для группы коммивояжеров в трехмерном пространстве для поиска оптимальных траекторий для каждого квадрокоптера и решение задачи синтеза управления квадрокоптером с целью обеспечения его устойчивого движения по заданным точкам пространственной траектории.

2. Разработан метод синтеза системы управления на основании нового метода символьной регрессии, метода вариационного аналитического программирования. Метод использует кодирование математических выражений методом аналитического программирования и метод малых вариаций кода базисного решения, заимствованный из метода сетевого оператора. Реализация алгоритма заключается в том, что разработчик задает на основе анализа задачи и опыта проектирования базисное решение и организует поиск на множестве малых вариаций базисного решения.

3. На основании исследования применения метода вариационного аналитического программирования для решения задачи синтеза системы управления установлено, что метод позволяет находить численное решение задачи синтеза системы управления и при этом имеет наиболее компактные коды записи математического выражения и вектора малых вариаций базисного решения. Метод позволяет кодировать математические выражения, включающие функции с любым количеством аргументов. Для применения метода вариационного аналитического программирования необходимо задать закодированное базисное решение и осуществить с помощью эволюционного генетического алгоритма поиск оптимального решения на основе принципа малых вариаций базисного решения.

4. С помощью вариационного генетического алгоритма и разработанного метода вариационного аналитического программирования решена задача мониторинга местности группой летающих роботов и решена задача синтеза

системы управления движением летающих роботов по пространственной траектории. Решением задачи синтеза является нелинейная обратная связь по всем переменным вектора состояния объекта управления.

4. Качество полученной системы автоматического управления летающим роботом исследовано с помощью моделирования. На основании анализа результатов моделирования установлено, что полученная система управления обеспечивает точное движение робота по пространственной траектории. Точность движения робота по траектории сохраняется для различных начальных условий.

СПИСОК ЛИТЕРАТУРЫ

1. Александров, В.В. Оптимальное управление движением. / В.В. Александров, В.Г. Болтянский, С.С. Лемак, Н.А. Парусников, В.М. Тихомиров. – М.: Физико-математическая наука, Наука/Интерпериодика. – 2005. – 376 с.
2. Алексеев, В.М. Оптимальное управление / В.М. Алексеев, В.М. Тихомиров, С.В. Фомин – М.: Наука. – 1979. – 384 с.
3. Антомонов, Ю.Г. Синтез оптимальных систем / Ю.Г. Антомонов – М.: Наукова думка. –1972. – 320 с.
4. Атиенсия Вильягомес, Х.М., Численный метод синтеза логико-функционального управления динамическим объектом/Х.М. Атиенсия Вильягомес, А.И. Дивеев // Современные проблемы науки и образования. – 2012. – № 3 URL: www.science-education.ru/103-6530.
5. Атиенсия Вильягомес, Х.М. Метод синтеза алгоритмов управления летающим роботом для системы точного земледелия Х.М. Атиенсия Вильягомес, А.И. Дивеев // Международный научный журнал. – 2013. – № 4. – С. 78–82.
6. Атиенсия Вильягомес, Х.М. Синтез интеллектуальной системы многоцелевого управления методом сетевого оператора / Х.М. Атиенсия Вильягомес, А.И. Дивеев // Вестник РУДН. Серия Инженерные исследования. – 2014.–№ 2. - С. 95–101.
7. Атиенсия Вильягомес, Х.М. Метод сетевого оператора для синтеза интеллектуальной системы управления динамическим объектом/ Х.М. Атиенсия Вильягомес, А.И. Дивеев // Электронный журнал «Cloud of Science». – 2014. – Том 1. – №2. –С. 191–198. – Режим доступа:<http://elibrary.ru/item.asp?id=22089435>.
8. Афанасьев, В.Н. Оптимальные системы управления / В.Н. Афанасьев – М.: Изд-во РУДН. –2007. – 260 с.

9. Афанасьев, В.Н. Управление неопределенными динамическими системами / В.Н. Афанасьев – М.: ФИЗМАТЛИТ. –2008. – 208 с.
10. Батищев, Д.И. Применение генетических алгоритмов к решению задач дискретной оптимизации / Д.И. Батищев, Е.А. Неймарк, Н.В. Старостин. – Нижний Новгород: Нижегородский государственный университет им. Н.И. Лобачевского. – 2007. – 88 с.
11. Болтянский, В.Г. Математические методы оптимального управления / В.Г. Болтянский – М.: Наука. –1968. – 408 с.
12. Галушкин, А.И. Нейронный сети: основы теории/ А.И. Галушкин. – М.: Горячая линия – Телеком. – 2010. – 496 с.
13. Гладков, Л.А. Генетические алгоритмы: Учебное пособие. 2-е изд./ Л.А. Гладков, В.В. Курейчик, В.М. Курейчик. –М: Физматлит. – 2006. – 320 с.
14. Гладков Л.А. Биоинспирированные методы в оптимизации / Л.А. Гладков, В.В. Курейчик, В.М. Курейчик и др. – М: Физматлит. – 2009. – 384 с.
15. Дивеев, А.И. Метод сетевого оператора / А.И. Дивеев. – М.: ВЦ РАН. –2010. – 178 с.
16. Дивеев, А.И. Метод сетевого оператора и его применение в задачах управления / А.И. Дивеев, Е.А. Софронова. – М.: РУДН. –2012. – 182 с.
17. Дивеев, А.И. Идентификация системы логического вывода методом сетевого оператора / А.И. Дивеев, Е.А. Софронова// Вестник РУДН. Серия Инженерные исследования. – 2010. – № 4. – С. 51–58.
18. Дивеев, А.И. Численный метод сетевого оператора для синтеза системы управления с неопределенными начальными значениями / А.И. Дивеев // Известия РАН. Теория и системы управления. – 2012. –№ 2. – С. 63–78.
19. Дивеев, А.И. Управление группой квадрокоптеров методом вариационного аналитического программирования / А.И. Дивеев, Н.Б. Конырбаев // Журнал «Труды МАИ»-2017. №96.
20. Дивеев, А.И. Численный метод вариационного аналитического программирования для структурно-параметрического синтеза системы

- управления /А.И. Дивеев, Н.Б. Конырбаев // Общероссийский научно-технический журнал «ПОЛЕТ». -2017. №11-12. -С. 31-37.
21. Дивеев, А.И. Эволюционный метод решения задачи группы квадрокоптеров для повышения качества мониторинга области /А.И. Дивеев, Н.Б. Конырбаев // Надежность и качество сложных систем. – 2017. № 4 (20). С. 64–72.
22. Дивеев, А.И. Синтез системы управления методом символьной регрессии / А.И. Дивеев, Н.Б. Конырбаев, С.И.Ибадулла // Сборник трудов международной научно-практической конференции, посвященной научно-педагогической деятельности академика А.Д.Тайманова «Современная математика: проблемы и приложения» - Алматы. 24-26 октября 2013г.
23. Дивеев, А.И. Вычислительный метод вариационного аналитического программирования для синтеза системы управления / А.И. Дивеев, Н.Б. Конырбаев // Современные проблемы науки и образования. – 2014. – № 2. – Режим доступа: <http://www.science-education.ru/116-12401>.
24. Дивеев, А.И. Метод вариационного аналитического программирования для решения проблемы синтеза системы управления / А.И. Дивеев, Н.Б. Конырбаев // Труды Международного симпозиума Надежность и качество; под ред. Н.К. Юркова. – Пенза - 26 мая – 01 июня. – 2014. – Т. 1. – С. 188–193.
25. Дивеев, А.И. Метод вариационного аналитического программирования для синтеза систем управления / А.И. Дивеев, Н.Б. Конырбаев // Труды одиннадцатого международного симпозиума Интеллектуальные системы INTELS'2014; под ред. К.А. Пупкова. – Москва. – 30 июня – 04 июля. –2014. – С. 78–83.
26. Дивеев, А.И. Синтез системы управления летающего робота методом вариационного аналитического программирования / А.И. Дивеев, Н.Б. Конырбаев // Научные технологии. №2-2015 г, т.16. С. 47-52.

27. Дивеев, А.И. Применение метода вариационного аналитического программирования для синтеза управления летающим роботом / А.И. Дивеев, Н.Б. Конырбаев // *Фундаментальные исследования*. – № 2015. № 3. – С. 51–57.
28. Дивеев, А.И. Исследование методов символьной регрессии для решения задач синтеза системы управления / А.И. Дивеев, М.В. Семиков // *Труды одиннадцатого международного симпозиума Интеллектуальные системы INTELS'2014*; под ред. К.А. Пупкова. – Москва. - 30 июня – 04 июля 2014 г. – С. 528–535.
29. Дивеев, А.И. Метод вариационного аналитического программирования для синтеза системы управления мобильным роботом / А.И. Дивеев, Н.Б. Конырбаев, В.И. Кравченя // *Труды VII Международной научно-практической конференции «Инженерные системы - 2014»*. Москва, 16 М 18 апреля 2014 г.; под общ. ред. Н.К. Пономарева. – М.: РУДН. –2014. – 134 с.
30. Дивеев, А.И. Вариационные методы символьной регрессии для задач управления и идентификации / А.И. Дивеев // *Труды X Международной конференции «Идентификация систем и задачи управления» SICPRO '15*. – Москва 26 – 29 янв., 2015 г. – С. 141-148.
31. Дорф, Р. *Современные системы управления* / Р. Дорф, Р. Бишоп –М.: «Лаборатория базовых знаний». –2002. – 832 с.
32. Ибадулла С.И. *Решение задачи синтеза системы управления методом вариационного генетического программирования* / С.И. Ибадулла // *Диссертация на соискание учёной степени к.т.н.* Москва, 2015.
33. Зубов, В.И. *Лекции по теории управления* / В.И. Зубов – СПб.: Издательство «Лань». – 2009. – 496 с.
34. Зубов, Н.Е. *Синтез развязывающих законов управления угловым движением возвращаемого аппарата с посадочной твердотопливной двигательной установкой, обеспечивающих минимизацию времени*

- переходного процесса. / Н. Е. Зубов, А. В. Лапин, Е. А. Микрин // Известия РАН. Теория и системы управления. – 2010. – № 3. – С. 155–166.
35. Зубов, Н.Е. Синтез развязывающих законов стабилизации орбитальной ориентации космического аппарата. / Н.Е. Зубов, Е.А. Микрин, М.Ш. Мисриханов, В.Н. Рябченко // Известия РАН. Теория и системы управления. – 2012. – № 1. – С. 92–108.
36. Зубов, Н.Е. Синтез законов управления космическим аппаратом, обеспечивающим оптимальное размещение полюсов замкнутой системой управления. / Н.Е. Зубов, Е.А. Микрин, М.Ш. Мисриханов, В.Н. Рябченко // Известия РАН. Теория и системы управления. – 2012. – № 3. – С. 98–110.
37. Зубов, Н.Е. Ленточные формулы анализа и синтеза управляемых динамических ММО-систем. / Н.Е. Зубов, Е.А. Микрин, М.Ш. Мисриханов, В.Н. Рябченко // Вестник МГТУ им. Н. Э. Баумана. Сер. Приборостроение. – 2014. – № 3. – С. 3–16.
38. Каллан, Р. Основные концепции нейронных сетей/ Р. Каллан. – М.: Издательский дом «Вильямс». – 2003. – 288 с.
39. Колесников, А.А. Синергетические методы управления сложными системами. Теория системного анализа / А.А. Колесников – М.: КомКнига. – 2006. – 240 с.
40. Колесников, А.А. Синергетические методы управления сложными системами. Механические и электромеханические системы. / А.А. Колесников и др. – М.: КомКнига. – 2006. – 304 с.
41. Кондратьев, Г.В. Геометрическая теория синтеза оптимальных стационарных гладких систем управления / Г.В. Кондратьев – М.: ФИЗМАТЛИТ. – 2003. – 144 с.
42. Конырбаев, Н.Б. Метод вариационного аналитического программирования и его применение для синтеза оптимальной системы управления / Н.Б. Конырбаев // Фундаментальные проблемы системной безопасности: материалы школы-семинара молодых ученых, Елец, 20-22 ноября 2014. – Елец: ЕГУ им. И.А. Бунина. – 2014. – С. 104–108.

- 43.Конырбаев, Н.Б. Сравнение метода вариационного аналитического программирования и метода сетевого оператора для управления колесным роботом /Н.Б.Конырбаев //Труды V Республиканской научно-практической конференции Образование. Наука. Инновация: Актуальные проблемы и пути развития. – Казахстан, Кызылорда. – 2014.
- 44.Кормен Т. Алгоритмы. Построение и анализ / Т. Кормен, Ч. Лайзерсон, Р.Ривест, К.Штайн // СПб.: Издательский дом «Вильямс», 2013, 1323 с.
- 45.Красовский, А.А. Системы управления полетом и их аналитическое конструирование / А.А. Красовский. – М.: Наука. – 1973. – 558 с.
- 46.Крутько, П.Д. Обратные задачи динамики управляемых систем. Нелинейные модели / П.Д. Крутько. М.: Наука, 1987. 304 с.
- 47.Кунцевич, В.М. Синтез систем автоматического управления с помощью функций Ляпунова / В.М. Кунцевич, М.М. Лычак – М.: Наука. – 1977. – 400 с.
- 48.Летов, А.М. Математическая теория процессов управления / А.М. Летов. – М.: Наука. –1981. – 255 с.
- 49.Ли, Э.Б. Основы теории оптимального управления / Э.Б. Ли, Л. Маркус. – М.: Наука. –1972. – 576 с.
- 50.Мерриэм, К. Теория оптимизации и расчет систем управления с обратной связью / К. Мерриэм. – М.: Мир. – 1967. – 548 с.
- 51.Меламед И.И. Задача коммивояжера. Точные методы / И.И. Меламед, С.И Сергеев, И.Х.Сигал //Автомат. и телемеханика, 1989, выпуск 10, С. 3–29.
- 52.Микрин, Е.А. Разработка моделей и методов проектирования информационно-управляющих систем космических аппаратов/ Е.А. Микрин, В.В. Кульба, Б.В. Павлов // Автоматика и телемеханика. – 2013. – № 3. – С. 38–50.
- 53.Орлов, С.А. Теория и практика языков программирования / С.А. Орлов. – СПб.: Питер. – 2013. – 688 с.
- 54.Пестерев,А.В. Синтез линеаризующего управления в задаче стабилизации движения автомобилеподобного робота вдоль криволинейного пути /

- А.В. Пестерев // Известия РАН. Теория и системы управления. –2013. – № 5. – С. 153–165.
55. Пупков, К.А. Методы синтеза оптимальных систем автоматического управления. / К.А. Пупков, Н.В. Фалдин, Н.Д. Егупов.–М.: Изд-во МГТУ им. Н.Э. Баумана. –2000.–512 с.
56. Рогачев, Г.Н. Генетическое программирование в задачах поиска системотехнических решений. / Г.Н. Рогачев, В.А. Егоров // 7-й Международный симпозиум «Интеллектуальные системы» (INTELS'2006) Краснодар, 26-30 июня 2006 г. – М.: РУСАКИ. – 2006. – С. 69–72.
57. Рогачев, Г.Н. Генетическое программирование в задачах поиска системотехнических решений / Г.Н. Рогачев // Вестник Самарского государственного технического университета. Сер. Технические науки. – 2006. –№ 40. – С. 37–42.
58. Ройтенберг, Я.Н. Автоматическое управление / Я.Н. Ройтенберг – М.: Наука. – 1978. – 552 с.
59. Сейдж, Э.П. Оптимальное управление системами. / Э.П. Сейдж, Ч.С. Уайт. – М.: Радио и связь. –1982.–392 с.
60. Справочник по теории автоматического управления; под ред. А.А. Красовского.– М.: Наука. –1987.–712 с.
61. Alnovani, G.H.A. Control Synthesis for Traffic Simulation in the Urban Road Network / G.H.A. Alnovani, A.I. Diveev, K.A. Pupkov, E.A. Sofronova // Preprints of the 18-th IFAC World Congress. Milan (Italy) August 28 – September 2, 2011. – P. 2196–2201.
62. Atiencia, V.J.M. The Network Operator Method for Synthesis of Intelligent Control System. / V.J.M. Atiencia, A.I. Diveev, E.A. Sofronova // Proceedings of the 2012 7th IEEE Conference on Industrial Electronics and Applications (ICIEA) 18-20 July, 2012, Singapore. –P. 169–174.
63. Bourmistrova, A. Control System Design Optimization via Genetic Programming / A. Bourmistrova, S. Khantsis. // Proceedings of CEC 2007. IEEE Congress on Evolutionary Computation. – 2007.–P. 1993–2000.

64. Bourmistrova, A. Genetic Programming in Application to Flight Control System Design Optimisation/ A. Bourmistrova, S. Khantsis. IN-TECH. –2010. – P. 195–228.
65. Davendra, D. 2010, Chaos driven evolutionary algorithms for the task of PID control/ D. Davendra., I. Zelinka., R.Senkerik // Computers & Mathematics with Applications. – Vol. 60. – No.4. –P. 1088–1104.
66. De Luca, A. Feedback Control of a Nonholonomic Car_Like Robot. / A. De Luca, G. Oriolo, C. Samson//Robot Motion Planning and Control;Ed. J.P. Laumond.– Springer. –1998. –P. 170–253.
67. Diveev A.I.
Variational Analytic Programming for Synthesis of Optimal Control for Flying Robot /A.I. Diveyev, N.B.Konyrbaev., S.I.Ibadulla., E.Yu.Shmalko// Preprints of the 11th IFAC Symposium on Robot Control, Salvador, BA, Brazil, August 26-28, 2015.-PP. 82-87.
68. Diveev A.I. Synthesis of Control for Group of Quadrotors in Task of Area Monitoring/A.I. Diveyev, N.B.Konyrbaev., S.I.Ibadulla., E.Yu.Shmalko// 2017 11th IEEE International Conference on Application of Information and Communication Technologies (AICT) 20-22 September 2017, Moscow, Russia.- PP. 365-370.
69. Diveyev, A.I. Application of network operator method for synthesis of optimal structure and parameters of automatic control system / A.I. Diveyev, E.A. Sofronova // Proceedings of 17-th IFAC World Congress. Seoul, 05.07.2008 – 12.07.2008. – P. 6106 – 6113.
70. Diveev, A.I. Numerical method of network operator for multi-objective synthesis of optimal control system. / A.I. Diveev, E.A. Sofronova // Proceedings of Seventh International Conference on Control and Automation (ICCA'09) Christchurch, New Zealand, December 9-11. – 2009. –P. 701-708.
71. Diveev, A.I. The Network Operator Method for Search of the Most Suitable Mathematical Equation / A.I. Diveev, E.A. Sofronova // Chapter in the book Bio-

- Inspired Computational Algorithms and Their Applications; Edited by Shangce Gao-. Intech, 2012. – February. – Croatia. – P. 19–42.
72. Diveev, A.I. Symbolic Regression Methods for Control System Synthesis. / A.I. Diveev, D.E. Kazaryan, E.A. Sofronova // 22nd Mediterranean Conference on Control and Automation (MED) University of Palermo. – June 16–19. –2014. Palermo, Italy. – P. 587–592.
73. Diveev A.I. Small Variations of Basic Solution Method for Non-numerical Optimization// IFAC-PapersOnLine, 2015. - Vol. 48 (25), P. 028-033.
74. Dorigo M. Ant Colony Optimization / M. Dorigo, T. Stützle // A Bradford Book Cambridge, Massachusetts, London, England: A Bradford Book, The MIT Press. 2004. 305 p.
75. Guerrero, J.A. Flight Formation Control/ J.A. Guerrero, R. Lozano. – John Wiley & Sons, Inc.: Hoboken, NJ 07030, USA.– 2012. – ISTE Ltd: London SW194, EU, UK.– 2012.– 321 p.
76. Koza, J.R. Genetic Programming: On the programming of computers by means of natural selection/ J.R. Koza. –Cambridge, Massachusetts: MIT Press.– 1992. – 819 p.
77. Koza, J.R. Automatic creation of human-competitive programs and controllers by means of genetic programming/ J.R. Koza, M.A. Keane, J. Yu, F.H. Bennett III, W. Mydlowec. // Genetic Programming and Evolvable Machines. – 2000. – Vol. 1.– No 1.–P. 121–164.
78. Koza, J.R., Performance improvement of machine learning via automatic discovery of facilitating functions as applied to a problem of symbolic system identification / Koza J. R., Keane M. A., Rice J. P. //IEEE International Conference on Neural Networks I. 1993. San Francisco, USA. 1993. P. 191 - 198.
79. Potvin J.V. "Genetic Algorithms for the Traveling Salesman Problem", Annals of Operations Research 1996, 63, P. 339-370.
80. Robbins, H. Junction phenomena for optimal control with state variable inequality constraints of third order/H. Robbins// Journal of Optimization Theory and Applications. – 1980. – No 31. – Vol. 1. – P. 85–99.

81. Ryan, C. Grammatical evolution: Evolving programs for an arbitrary language. / C. Ryan, J.J. Collins, M.O. Neill // EuroGP. 1998: LNCS.–Vol. 1391.– P. 83–95.
82. Ryan, C. How to do anything with Grammars/ C. Ryan, M. O'Neill // Proc. of the Bird of a Feather Workshops, Genetic and Evolutionary Computation Conference 2002.– P. 116–119.
83. Zelinka, I. Analytic programming by Means of Soma Algorithm/ I. Zelinka // In Proc. 8th International Conference on Soft Computing Mendel'02, Brno, Czech Republic. – 2002. – P. 93–101.
84. Zelinka, I. Analytic programming / I. Zelinka, Z. Oplatkova // Comparative Study. CIRAS'03, The second International Conference on Computational Intelligence, Robotics, and Autonomous Systems, Singapore. – 2003.
85. Zelinka, I. Analytic Programming —Symbolic Regression by Means of Arbitrary Evolutionary Algorithms / I. Zelinka, L. Nolle, Z. Oplatkova // Journal of Simulation. –2012. – Vol. 6, – No 9. – P. 44–56.
86. Zelinka, I., Analytical Programming – a Novel Approach for Evolutionary Synthesis of Symbolic Structures Chapter in the book Evolutionary Algorithms Edited by Prof. / I. Zelinka. D. Davendra, R. Senkerik, R. Jasek and Z. Oplatkova // EISUKA Intech. Printed/ – 2011. – April, Croatia. – P. 149-176.
87. Zelinka I. Synthesis of feedback controller for three selected chaotic systems by means of evolutionary techniques/I. Zelinka., R. Senkerik., Z. Oplatkova., D. Davendra // Analytic programming, Mathematical and Computer Modelling. – Vol. 57, – No.1–2. –P. 57–67.