

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

На правах рукописи

Ибадулла Сабит Ибадуллаулы

**РЕШЕНИЕ ЗАДАЧИ СИНТЕЗА СИСТЕМЫ УПРАВЛЕНИЯ
МЕТОДОМ ВАРИАЦИОННОГО ГЕНЕТИЧЕСКОГО
ПРОГРАММИРОВАНИЯ**

05.13.01 – Системный анализ, управления и обработка информации
(промышленность)

Диссертация

на соискание ученой степени кандидата технических наук

Научный руководитель:
д.т.н., профессор А.И. Дивеев

Москва, 2014г.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	3
1. ЗАДАЧА СИНТЕЗА СИСТЕМЫ УПРАВЛЕНИЯ.....	16
1.1. ФОРМАЛЬНАЯ ПОСТАНОВКА ЗАДАЧИ СИНТЕЗА УПРАВЛЕНИЯ.....	16
1.2. МНОГОКРИТЕРИАЛЬНАЯ ЗАДАЧА ЧИСЛЕННОГО СИНТЕЗА УПРАВЛЕНИЯ.....	22
1.3. МЕТОД СЕТЕВОГО ОПЕРАТОРА.....	24
1.4. ВЫВОДЫ К ГЛАВЕ 1.....	32
2. МЕТОД ВАРИАЦИОННОГО ГЕНЕТИЧЕСКОГО ПРОГРАММИРОВАНИЯ.....	33
2.1. МЕТОД ГЕНЕТИЧЕСКОГО ПРОГРАММИРОВАНИЯ.....	34
2.2 КОДЫ СИМВОЛОВ МЕТОДА ГЕНЕТИЧЕСКОГО ПРОГРАММИРОВАНИЯ.....	39
2.3 МЕТОД ВАРИАЦИОННОГО ГЕНЕТИЧЕСКОГО ПРОГРАММИРОВАНИЯ.....	48
2.4 ПРИМЕР СИНТЕЗА УПРАВЛЕНИЯ МЕТОДАМИ ГЕНЕТИЧЕСКОГО ПРОГРАММИРОВАНИЯ И ВАРИАЦИОННОГО ГЕНЕТИЧЕСКОГО ПРОГРАММИРОВАНИЯ.....	59
2.5 ВЫВОДЫ К ГЛАВЕ 2.....	64
3. СИНТЕЗ СИСТЕМЫ УПРАВЛЕНИЯ МОБИЛЬНЫМ РОБОТОМ МЕТОДОМ ВАРИАЦИОННОГО ГЕНЕТИЧЕСКОГО ПРОГРАММИРОВАНИЯ.....	65
3.1 МАТЕМАТИЧЕСКАЯ МОДЕЛЬ ОБЪЕКТА УПРАВЛЕНИЯ.....	66
3.2 СИНТЕЗ СИСТЕМЫ УПРАВЛЕНИЯ ДВИЖЕНИЕМ МОБИЛЬНОГО РОБОТА ПО ПРОСТРАНСТВЕННЫМ ТРАЕКТОРИЯМ.....	78
3.2.1 Задача синтеза управления движением вдоль пространственной траектории.....	79
3.2.2 Пример синтеза системы управления движением по траектории.....	81
3.2 ВЫВОДЫ К ГЛАВЕ 3.....	85
ЗАКЛЮЧЕНИЕ.....	86
СПИСОК СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ.....	88
ЛИТЕРАТУРА.....	89
ПРИЛОЖЕНИЕ.....	95

ВВЕДЕНИЕ

Работа посвящена решению задачи синтеза системы управления. В рамках математической формулировки задача синтеза управления состоит в нахождении многомерной функции, определяющей по значениям компонент вектора состояния объекта управления такие значения компонент вектора управления, которые обеспечивают достижение цели управления при оптимальном значении критерия качества управления.

С позиции технической формулировки задача синтеза системы управления заключается в создании блока управления, работающего на основе реализации принципа обратной связи. Блок управления должен получать с датчиков сигналы, по которым возможно определение состояния объекта управления, и вырабатывать сигналы, по которым создаются управляющие воздействия, обеспечивающие объектом достижение цели управления с минимальным значением некоторого показателя качества.

В теории управления регуляторы, использующие сигналы обратной связи, давно и успешно используются для повышения качества управления и улучшения свойств объекта [17, 30, 37, 40, 43]. Такой способ регулирования, например, использует широко известный пропорционально интегрально дифференцирующий регулятор (ПИД-регулятор). Для систем регулирования не ставится задача целевого управления, например, перемещения объекта из одного состояния в другое. Для этой цели предполагается использование внешнего управляющего сигнала, который поступает на объект управления, а система регулирования, при этом обеспечивает качество результата воздействия управляющего сигнала на объект, например, уменьшает время переходного процесса или величину перерегулирования.

В задачах синтеза управления необходимо найти обратную связь, которая обеспечивает выработку управляющего внешнего сигнала по состоянию объекта, при этом объект перемещается в заданное целевое состояние, т.е. решается задача достижения цели управления, а величина некоторой оценки качества движения объекта в пространстве состояний принимает минимально возможное значение.

Например, такой величиной оценки качества управления может быть минимальное время перемещения объекта в целевое состояние или интегральная оценка квадрата управляющего воздействия.

Основным способом решения задачи управления, обеспечивающего достижение цели с минимизацией критерия качества, сегодня является оптимальное программное управление [1, 2, 5, 32]. Задача оптимального управления также включает цель управления и критерий качества управления, который необходимо минимизировать. В отличие от задачи синтеза управления в задаче оптимального управления для объекта строго заданы начальные условия, или они находятся в процессе решения задачи из условий трансверсальности. В результате решения задачи оптимального управления находят управление как функцию времени, т.е. функцию, не зависящую от состояния объекта управления. При реализации такого управления на реальном объекте, как правило, в начале определяют с помощью моделирования оптимальную траекторию движения объекта в пространстве состояний, а затем с помощью систем регулирования обеспечивают движение объекта в окрестности заданной оптимальной траектории.

В отличие от программного управления, реализуемого в результате решения задачи оптимального управления, в задаче синтеза системы управления начальные условия заданы ни одной точкой в пространстве состояний, а в виде области или множества точек в пространстве состояний. Полученная в результате решения задачи синтеза синтезирующая функция обеспечивает оптимальное перемещение объекта управления из любого начального состояния из данного множества в терминальное целевое состояние. Если начальные условия в задачах оптимального управления и синтеза управления совпадают, то полученная синтезирующая функция дает те же значения управления во времени, что и оптимальное программное управление, полученное в результате решения задачи оптимального управления. В этом смысле решение задачи синтеза управления эквивалентно решению множества задач оптимального управления.

Современный этап развития технологии характеризуется большим количеством полуавтоматических или автоматизированных систем управления, в которых функцию блока управления, обеспечивающего достижения цели управления и качества, выполняет человек-оператор, водитель, пилот и т.п. Такое управление наблюдается и в широко распространенных сегодня роботизированных устройствах. Человек - оператор сам с помощью органов чувств, прежде всего зрения, и приборов, указывающих параметры состояния объекта управления, выполняет функцию обратной связи и вырабатывает управляющие сигналы на объект для достижения им цели управления и обеспечения качества. В данной интерпретации задача синтеза системы управления заключается в замене человека оператора автоматическим устройством, которое вырабатывает управляющие воздействия по сигналам, поступающим с приборов, определяющих состояние объекта управления. Сегодня автоматически управляемых объектов без использования человека-оператора в контуре управления не очень много, но очевидно, что с каждым днем их становится все больше и больше.

Методы создания блоков автоматического управления или решения задачи синтеза системы управления можно разделить на два класса. К первому классу, сегодня наиболее используемому, следует отнести инженерные методы синтеза [20, 31, 42]. Эти методы включают детальное подробное исследование объекта управления и самой задачи, которая ставится перед системой управления. Далее на основе опыта разработчика, его интуиции и результатов анализа экспериментальных данных конструируется структура системы управления. Затем в этой системе производится настройка ее параметров, возможно с применением численных методов оптимизации.

Ко второму классу методов для решения задачи синтеза системы управления следует отнести математические или численные методы. Эти методы рассматривают только математическую модель объекта управления. Структура и параметры синтезированной системы управления должны получаться в результате применения этих методов к рассматриваемой математической модели

объекта управления. Математические методы синтеза не являются альтернативой инженерных методов, а скорее предназначены в помощь им для замены интуитивных решений математически обоснованными или найденными с помощью численных методов структурами и параметрами систем управления.

Диссертация посвящена разработке нового численного метода вариационного генетического программирования для решения задачи синтеза системы управления.

Математические методы решения задачи синтеза управления делятся на два подкласса. Это методы, основанные на использовании аналитических подходов к решению задачи синтеза [6, 23, 24, 26, 33-36,] и, непосредственно, численные методы, ориентированные на машинном поиске решения без предварительных аналитических преобразований.

В аналитических методах решения задачи синтеза управления наиболее сильными результатами являются: уравнение Беллмана [5, 40, 41] и решение задачи аналитического конструирования оптимальных регуляторов (АКОР) [3, 4, 25, 28].

Уравнение Беллмана получено на основе предположения о том, что правые части системы дифференциальных уравнений, описывающих модель объекта управления, дифференцируемы по всем своим аргументам, и искомая синтезирующая функция также дифференцируема по аргументам. В результате уравнение Беллмана представляет собой нелинейное дифференциальное уравнение в частных производных, в общем случае не имеющее аналитического решения. Решение задачи АКОР – это решение одного частного случая уравнения Беллмана, когда модель объекта управления описывается линейными дифференциальными уравнениями, величина управления не ограничена и функционал качества описывается квадратичными уравнениями. При этом решение задачи АКОР получается интуитивно из предположения, что функция Беллмана представляет собой квадратичную форму от векторов пространства состояний [18, 26-28].

Решение задачи АКОР было получено в 60-е годы. На основании решения этой задачи строится система стабилизации объекта относительно одного заданного состояния, при этом система стабилизации реализует линейную многомерную функцию относительно координат пространства состояний объекта управления. Дальнейшие исследования показали, что квадратичная функция Беллмана совпадает с функцией Ляпунова, если рассматривать замкнутую модель объекта управления и требовать от описывающей ее системы дифференциальных уравнений обеспечения свойства устойчивости по Ляпунову относительно заданного состояния.

Дальнейшие аналитические подходы к решению задачи синтеза управления были, в основном связаны с применением метода АКОР для различных задач, в том числе нелинейных, нестационарных, неопределенных или подверженных внешним возмущениям. Основным этапом применения метода АКОР является этап линеаризации системы относительно заданного терминального состояния. Заметим, что применение метода АКОР приводит к созданию линейной синтезирующей функции, поэтому можно сразу предположить, что данный подход – это попытка поиска решения для задачи синтеза управления в рамках систем с заданной структурой. Следовательно, метод АКОР сводит задачу структурно параметрического синтеза системы управления к задаче параметрического синтеза. Современные вычислительные методы позволяют сегодня эффективно находить оптимальные значения параметров на пространстве достаточно большой размерности без применения аналитических преобразований.

Проблема создания метода для решения задачи синтеза управления заключается, прежде всего, в нахождении метода поиска структуры функции, т.е. ее аналитического описания. До начала 90-х годов прошлого века машинные методы поиска структуры функции отсутствовали. Представление функций в виде разложения в ряды является также сведением задачи поиска структуры к задаче поиска параметров.

Появившиеся в начале 60-х годов нейронные сети следует рассматривать как универсальный метод аппроксимации любых функций, тем более, после

доказательства теоремы Колмогорова о том, что любая функция представима в виде сумм ограниченного числа элементарных функций. Следует заметить, что нейронные сети – это функции с заданной, хотя и регулярно наращиваемой, структурой, поэтому использование нейронных сетей для решения задачи синтеза управления – это также сведение задачи структурно параметрического синтеза к параметрическому синтезу, пусть с очень большим количеством параметров. Несмотря на огромную популярность метода нейронных сетей для различных задач, работ по решению задач синтеза систем управления методами нейронных сетей в мировой литературе практически нет. Для убедительности достаточно посмотреть последние конгрессы, проводимые международной федерацией по автоматическому управлению (International Federation of Automatic Control, IFAC) в 2008 году в Сеуле, в 2011 году в Милане, в 2014 году в Кейптауне. В каждом из конгрессов было представлено около 3000 статей, но работы по решению задачи синтеза системы управления методами нейронных сетей отсутствуют. Не было работ по синтезу системы управления методом нейронных сетей и на юбилейной конференции (Всероссийскому совещанию по управлению, ВСПУ-2014), посвященной 75-летию Института Проблем Управления им. В.А. Трапезникова, на которой было представлено 1074 доклада. В основном нейронные сети в управлении используются в адаптивных системах для настройки параметров при изменении модели объекта или режимов функционирования.

Основным научным результатом, который дал импульс к созданию численных методов для решения задачи синтеза системы управления, является метод генетического программирования, разработанный профессором университета Стэнфорда Дж. Козой в 1992 году [53, 54]. Метод был предназначен для решения задачи автоматического написания программ. Метод использует универсальную для описания строк программы структуру данных, польскую запись в форме бесскобочной последовательности символов, и эволюционный алгоритм поиска, генетический алгоритм. Метод генетического программирования сегодня очень популярен и применяется для решения большого количества разнообразных задач. Достаточно посмотреть труды

ежегодных конференций по генетическим и эволюционным вычислениям (Genetic and Evolutionary Computing Conference GECCO) или европейской конференции по генетическому программированию (Europe Genetic Programming, EuroGP).

Поскольку искомая в задаче синтеза системы управления функция должна вычисляться с помощью программно-реализуемого алгоритма, то метод генетического программирования, как метод поиска структуры функции, применим для решения задачи синтеза системы управления. Исследования по использованию метода генетического программирования для решения задачи синтеза систем управления проводятся с момента его появления [38, 39, 45, 46, 54]. Следует отметить, что применение генетического программирования к решению задачи синтеза требует больших вычислительных мощностей, так как рекомендуемое множество возможных решений должно содержать не менее ста тысяч элементов.

Проблема использования метода генетического программирования для решения задачи синтеза системы управления заключается в наличии жестких требований, которые выдвигаются к возможным решениям, это выполнение краевых условий, описывающих цель управления, минимизация критерия качества, и выполнение, как правило, неформального требования по обеспечению устойчивости замкнутой системы управления. При экспериментальном исследовании метода генетического программирования было установлено, что подавляющее большинство элементов во множестве случайно сгенерированных возможных решений не удовлетворяют заданным требованиям, поэтому поиск решения и должен осуществляться на достаточно большом множестве. Вторым недостатком генетического программирования является процесс декодирования искомой функции по строке записи. Для получения математического выражения из кода польской записи необходимо наличие лексического анализатора, который переводит коды символов в коды соответствующих функций. Третьим недостатком метода генетического программирования является сложность выполнения операции скрещивания, которая осуществляется только в точках, где символы описывают функции с одинаковым количеством аргументов, а

выполнение операции скрещивания заключается с помощью достаточно сложного алгоритма обмена подстрок разной длины.

Для усовершенствования метода генетического программирования с целью его применения к решению задачи синтеза управления в 2006 году профессором А.И. Дивеевым из Вычислительного центра им. А.А. Дородницына Российской академии наук был разработан метод сетевого оператора [7, 8, 16, 49-51]. Метод позволяет представлять математические выражения в виде ориентированного графа и записывать этот граф в память вычислительной машины в форме целочисленной матрицы. Существенными преимуществами метода сетевого оператора являются отсутствие лексического анализатора при вычислении математического выражения по матрице сетевого оператора и использование при поиске принципа малых вариаций базисного решения.

Множество возможных решений в методе сетевого оператора генерируют с помощью одного базисного решения и множества наборов одинаковой длины из векторов вариаций. Базисное решение задается на основе анализа решаемой задачи. При формировании базисного решения проектировщик использует собственный опыт, теоретические исследования и интуитивные предположения. Чем лучше задано базисное решение, тем быстрее алгоритм находит решение задачи синтеза управления. Каждый вектор вариации позволяет осуществить одну малую вариацию матрицы сетевого оператора базисного решения. Генетическая операция скрещивания осуществляется классическим способом в любой точке двух отобранных наборов с помощью легкого алгоритма обмена оставшихся после точки скрещивания частей наборов из векторов вариаций.

С помощью метода сетевого оператора решено довольно большое количество задач синтеза управления, в том числе и достаточно сложных задач по синтезу управлений космическими и летательными аппаратами. К недостаткам метода сетевого оператора следует отнести ограниченный набор используемых функций. Метод включает только функции с одним или двумя аргументами, причем функции с двумя аргументами должны быть коммутативны, ассоциативны и иметь единичный элемент. Данное ограничение существенно при

синтезе интеллектуальных систем управления, которые сегодня часто востребованы при разработке робототехнических комплексов. Интеллектуальные системы управления, как правило, используют оператор условия (if), реализация которого в виде функции требует наличия трех аргументов. Для решения задач синтеза систем интеллектуального управления был разработан метод логического сетевого оператора [7-9, 44, 47], который использует логические функции с одним или двумя аргументами, однако данный оператор также не использует полноценную функцию, описывающую оператор условия.

В отличие от сетевого оператора метод генетического программирования может использовать функции без специальных свойств и с любым количеством аргументов. В процессе исследования численных методов синтеза возникла идея совместить достоинства метода сетевого оператора, принцип малых вариаций базисного решения, с достоинствами метода генетического программирования, использование любых функций с произвольным количеством аргументов.

В диссертационной работе поставлена задача решение и исследования нового вычислительного метода для синтеза системы управления. Применяемый метод для решения задачи синтеза системы управления, имеет преимущества обоих методов генетического программирования и сетевого оператора. Этот метод называется методом вариационного генетического программирования, и он, как и метод генетического программирования, включает любые функции с произвольным количеством аргументов, и при поиске решения использует принцип малых вариаций базисного решения.

Сегодня решение задачи синтеза управления необходимо для создания автоматических роботизированных устройств, мобильных роботов, беспилотных летательных аппаратов и др. Настоящий период развития техники позволяет использовать данные устройства в различных областях для решения разнообразных полезных обществу задач, при решении которых эти роботизированные устройства выполняют сложные функции. Большинство таких устройств работает под управлением человека-оператора. Современный этап развития технологии характеризуется переходом режимов управления

роботизированными устройствами от управления человеком-оператором к автоматическому автономному режиму функционирования. Для создания систем автоматического управления необходимо решить задачу синтеза управления. Аналитические методы синтеза управления работают только для несложных моделей объектов управления. Вычислительные методы синтеза разработаны только для эффективного поиска значений оптимальных параметров. Методы синтеза структур систем управления требуют больших вычислительных затрат. Использование современных достижений в области алгоритмизации позволяет сегодня построить эффективные вычислительные методы синтеза систем управления на основе эволюционных алгоритмов поиска. Существующие методы синтеза систем управления, построенные на основе метода сетевого оператора и генетического программирования обладают определенными недостатками. Метод сетевого оператора использует ограниченный набор функций, а метод генетического программирования требует больших вычислительных ресурсов.

Диссертация посвящена решению необходимой для современного этапа развития технологии задачи, а именно применения эффективного вычислительного метода для синтеза системы управления. Диссертация посвящена решению востребованной обществом задачи, которая сегодня не решена. Этим объясняется актуальность темы диссертационной работы.

Применение нового вычислительного метода синтеза системы управления опирается на уже существующие методы. Применяемый метод строится на основе эволюционного поиска и использует преимущества известных методов сетевого оператора и генетического программирования. И этот метод использует принцип малых вариаций базисного решения из метода сетевого оператора и структуру данных в виде символьной последовательности аналогичную методу генетического программирования.

Использование последних достижений в области алгоритмизации и известных вычислительных методов синтеза систем управления определяет высокую степень разработанности используемых в диссертации подходов.

Объект исследования

Объектом исследования является система управления мобильным роботом, модель которого описывается системой ОДУ.

Положения, выносимые на защиту

- 1) Решение задачи синтеза системы управления новым вычислительным методом вариационного генетического программирования для синтеза системы управления, построенным на основе методов сетевого оператора и генетического программирования;
- 2) Вычислительный эволюционный алгоритм, использующий принцип малых вариаций базисного решения для метода генетического программирования;
- 3) Решение задачи синтеза системы управления мобильным роботом с помощью метода вариационного генетического программирования.
- 4) Комплекс программ, реализующий вычислительный метод для решения задачи синтеза системы управления мобильным роботом.

Цель и задачи диссертационной работы

Целью диссертационного исследования заключается в применении нового вычислительного метода вариационного генетического программирования для решения задачи синтеза системы управления.

Для достижения поставленной цели в диссертации необходимо решить следующие задачи:

1. разработать алгоритм решения задачи синтеза управления вычислительным методом вариационного генетического программирования, построенным на основе методов сетевого оператора и генетического программирования;
2. разработать эволюционный алгоритм поиска оптимального решения на основе принципа малых вариаций базисного решения в методе генетического программирования;
3. решить задачу синтеза системы управления мобильным роботом с помощью метода вариационного генетического программирования.

Научная новизна

диссертационной работы определяется решением задачи синтеза системы управления вычислительным методом вариационного генетического программирования.

диссертационной работы определяется созданием нового вычислительного метода вариационного генетического программирования для решения задачи синтеза системы управления.

Теоретическая значимость работы

заключается в анализе работоспособности алгоритма для поиска решения задачи синтеза системы управления, построенного на основе принципа малых вариаций базисного решения в методе генетического программирования.

Практическая значимость работы

заключается в том, что применяемый новый вычислительный метод предназначен для решения синтеза систем управления и в диссертационной работе новый метод применен для решения задачи синтеза системы управления мобильным роботом.

Методология и методы исследования,

используемые в диссертационной работе, основываются на программной реализации в решении поставленной задачи с применением метода вариационного генетического программирования и моделировании получаемой в результате синтеза систем управления.

Предметом исследования

диссертационной работы является применение нового вычислительного метода вариационного генетического программирования для решения задачи синтеза системы управления.

Публикации

Основные положения исследования отражены в 9 публикациях автора, из которых 3 работы, опубликованы в журналах, рекомендуемых ВАК РФ. В совместных работах результаты принадлежат соавторам в равных долях.

Апробация работ

Основные положения исследования докладывались и обсуждались

- на международном симпозиуме «Надежность и качество» в г. Пензе в 2014 г.;

- труды одиннадцатого международного симпозиума «Интеллектуальные системы - INTELS'2014» в г. Москве в 2014 г.,

а также на семинарах кафедры Кибернетики и мехатроники РУДН и отдела Нелинейного анализа и проблем безопасности ВЦ РАН.

Структура и объем работы

Диссертация состоит из введения, трех глав, приложения и списка литературы. Диссертация содержит 87 страниц текста из 126, включает 30 рисунка, 2 таблиц. Список литературы содержит 54 наименования.

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 диссертационной работы рассмотрена формальная постановка задачи синтеза системы управления. Приведен обзор известных вычислительных методов синтеза систем управления. Представлено краткое описание метода сетевого оператора и используемого в нем принципа малых вариаций базисного решения.

Во второй главе приведен разработанный вычислительный метод вариационного генетического программирования для синтеза системы управления. Приведено описание метода генетического программирования. Описаны малые вариации структуры данных, используемой в методе генетического программирования, Описан эволюционный генетический алгоритм для решения задачи синтеза системы управления методом вариационного генетического программирования. Приведен сравнительный пример решения задачи синтеза системы управления для объекта, описываемого математической моделью «хищник-жертва» методами генетического программирования и вариационного генетического программирования.

В третьей главе приведен пример решения задачи синтеза системы управления движением мобильного робота с учетом пространственных ограничений. Приведен анализ эффективности разработанного метода. Представлено на основе моделирования исследование полученной в результате синтеза системы управления.

1. ЗАДАЧА СИНТЕЗА СИСТЕМЫ УПРАВЛЕНИЯ

В работе рассматривается задача синтеза системы управления. В задаче задана математическая модель объекта управления, заданная в виде системы обыкновенных дифференциальных уравнений. Дифференциальные уравнения должны описывать динамику объекта управления и включать в правые части дифференциальных уравнений ограниченный по величине вектор управления. Для модели объекта управления задана область начальных условий. Задан в виде функционала критерий качества управления. Заданы в форме многообразия терминальные условия, к которым управление должно привести объект за конечное время. Необходимо найти управление в виде системы нелинейных уравнений, описывающих функциональную зависимость значения вектора управления от значений вектора состояния объекта управления.

После подстановки найденного математического выражения в модель объекта управления получаем замкнутую систему дифференциальных уравнений, в правые части которой не входит вектор управления. Любое частное решение, полученное в результате решения задачи синтеза, замкнутой системы дифференциальных уравнений с начальными условиями из заданной области должно в определенный момент времени удовлетворять уравнениям, описывающим терминальное состояние, и значение функционала, вычисленное вдоль всего решения, должно иметь минимально возможное значение.

1.1 Формальная постановка задачи синтеза управления

Приведем формальное описание постановки задачи.

Задана математическая модель объекта управления

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

где \mathbf{x} - вектор состояния объекта управления, $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{x} = [x_1 \dots x_n]^T$, \mathbf{u} - вектор управления, $\mathbf{u} \in U \subseteq \mathbb{R}^m$, $\mathbf{u} = [u_1 \dots u_m]^T$, U - замкнутое ограниченное множество, $m \leq n$, $\mathbf{f}(\mathbf{x}, \mathbf{u}): \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$, $\mathbf{f}(\mathbf{x}, \mathbf{u}) = [f_1(\mathbf{x}, \mathbf{u}) \dots f_n(\mathbf{x}, \mathbf{u})]^T$.

Задана область начальных значений

$$X_0 \subseteq \mathbb{R}^n, \quad (1.2)$$

где X_0 - замкнутое ограниченное множество.

Заданы терминальные условия

$$\varphi_i(\mathbf{x}(t_f)) = 0, \quad i = \overline{1, l}, \quad (1.3)$$

где $l \leq n$, $t_f \leq t^+$, t^+ - заданное предельное время процесса управления, $0 < t^+ < \infty$.

Задан функционал качества

$$J = \int_0^{t_f} f_0(\mathbf{x}, \mathbf{u}) dt \rightarrow \min, \quad (1.4)$$

где величина t_f определяется выполнением условия (1.3), функция $f_0(\mathbf{x}, \mathbf{u})$ ограничена снизу, $f_0(\mathbf{x}, \mathbf{u}): \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^1$.

Необходимо найти управление в виде

$$\mathbf{u} = \mathbf{h}(\mathbf{x}), \quad (1.5)$$

где функция $\mathbf{h}(\mathbf{x}): \mathbb{R}^n \rightarrow \mathbb{R}^m$, $\mathbf{h}(\mathbf{x}) = [h_1(\mathbf{x}) \dots h_m(\mathbf{x})]^T$, обеспечивает для частного решения $\tilde{\mathbf{x}}(t, \mathbf{x}^0)$ замкнутой системы дифференциальных уравнений $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{h}(\mathbf{x}))$, вычисленного для начального значения из заданной области (1.2), $\mathbf{x}(0) = \mathbf{x}^0 \in X_0$, следующие свойства:

а) выполняются терминальные условия (1.3), т.е. $\exists \tilde{t}_f < t^+$ такое, что

$$\varphi_i(\tilde{\mathbf{x}}(\tilde{t}_f, \mathbf{x}^0)) = 0, \quad i = \overline{1, l};$$

б) выполняются ограничения на управление $\mathbf{h}(\tilde{\mathbf{x}}(t, \mathbf{x}^0)) \in U, 0 \leq t \leq \tilde{t}_f$;

в) значение функционала (1.4), вычисленного для полученного решения принимает минимальную величину

$$\int_0^{\tilde{t}_f} f_0(\tilde{\mathbf{x}}(t, \mathbf{x}^0), \mathbf{h}(\tilde{\mathbf{x}}(t, \mathbf{x}^0))) dt = \min_{\mathbf{u}(t) \in U} \int_0^{t_f} f_0(\mathbf{x}(t, \mathbf{x}^0), \mathbf{u}(t)) dt, \quad (1.6)$$

где $\mathbf{x}(t, \mathbf{x}^0)$ - решение системы уравнений $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}(t))$ для тех же начальных условий $\mathbf{x}(0) = \mathbf{x}^0 \in X_0$ и управления $\mathbf{u}(t) \in U$, которое подобрано так, чтобы решение в момент $t_f < t^+$, то же удовлетворяет терминальным условиям (1.3) $\varphi_i(\mathbf{x}(t_f, \mathbf{x}^0)) = 0, i = \overline{1, l}$, при этом t_f не обязательно равно \tilde{t}_f .

Часто в качестве терминальных условий или цели управления (1.3) в общей постановке задачи синтеза используют многообразие размерности ноль, $n - l = 0$, или попадание в одну точку пространства состояния, $\mathbf{x}(t_f) = \mathbf{x}^f$. В качестве области начальных условий (1.2) используют все пространство состояний, $X_0 = \mathbf{R}^n$. В этом случае задача синтеза системы управления соответствует задаче стабилизации объекта управления относительно терминального состояния.

Аналитических методов решения задачи синтеза управления в общей постановке нет. Для решения задачи стабилизации, $\mathbf{x}(t_f) = \mathbf{x}^f$ и при условии дифференцируемости правых частей системы дифференциальных уравнений (1.1) модели объекта управления по своим аргументам получено уравнение Беллмана [5, 40], которому должно удовлетворять оптимальное управление

$$\min_{\mathbf{u} \in U} \left\{ f_0(\mathbf{x}, \mathbf{u}) + \left(\frac{\partial \mu(\mathbf{x})}{\partial \mathbf{x}} \right)^T \mathbf{f}(\mathbf{x}, \mathbf{u}) \right\} = 0. \quad (1.7)$$

Решение уравнение (1.7) получено для моделей объектов управления, описываемых системой линейных дифференциальных уравнений

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}, \quad (1.8)$$

где \mathbf{A} и \mathbf{B} - числовые матрицы размерностей $n \times n$ и $n \times m$, соответственно, с квадратичным критерием качества

$$J = \int_0^{t_f} (\mathbf{x}^T \mathbf{N} \mathbf{x} + q \mathbf{u}^T \mathbf{u}) dx \quad (1.9)$$

где \mathbf{N} - положительно определенная матрица размерности $n \times n$, q весовой коэффициент, $q > 0$, и нулевыми терминальными условиями

$$\mathbf{x}(t_f) = 0. \quad (1.10)$$

Решение дифференциального уравнения в частных производных (1.7) искалось из предположения, что неизвестная функция $\mu(\mathbf{x})$ описывается квадратичной формой

$$\mu(\mathbf{x}) = \mathbf{x}^T \mathbf{P} \mathbf{x}, \quad (1.11)$$

где \mathbf{P} - матрица размерности $n \times n$.

Решение получается в виде линейной зависимости управления от координат пространства состояний

$$\mathbf{u} = \frac{1}{2q} \mathbf{B}^T \mathbf{P} \mathbf{x}. \quad (1.12)$$

Данный результат известен как метод аналитического конструирования оптимальных регуляторов (АКОР) [3, 4, 28].

Заметим, что оптимальная структура управления (1.12) требует нахождения элементов числовой матрицы размерности $m \times n$

$$\mathbf{u} = \mathbf{Q} \mathbf{x}, \quad (1.13)$$

где $\mathbf{Q} = [q_{i,j}]$, $q_{i,j} \in \mathbb{R}^1$, $i = \overline{1, m}$, $j = \overline{1, n}$.

Данная задача решается даже при достаточно больших значениях m и n на современных вычислительных средствах с помощью эволюционных алгоритмов или другими вычислительными методами глобальной оптимизации.

Другим аналитическим методом к решению задачи синтеза управления является подход, основанный на использовании функций Ляпунова [18, 27]. В начале 60-х годов Н.Н. Крассовским было установлено, что принципу оптимальности и уравнению Беллмана соответствуют только такие функции, которые являются функциями Ляпунова для замкнутой системы. Следовательно, найденное с помощью функций Ляпунова управление не только обеспечивает устойчивость движения, но является оптимальным.

Данные аналитические методы решают задачу стабилизации, используют функционалы определенного и линейные или линеаризованные относительно положений устойчивости модели объектов управления. В подавляющем большинстве случаев полученная в результате решения таких задач синтеза структура системы управления является линейной функцией вида (1.13) от координат пространства состояний. Как было сказано выше, для решения этих задач достаточно использовать эффективный вычислительный алгоритм для поиска коэффициентов числовой матрицы без аналитических преобразований.

В 90-х годах прошлого века Колесниковым А.А. был разработан метод аналитического конструирования агрегированных регуляторов (АКАР), который почему-то называется также синергетическим методом синтеза [23, 42]. Для реализации данного метода синтеза разработчику необходимо знать или предполагать оптимальное движение замкнутого объекта управления в пространстве состояний. Данное движение разработчик описывает с помощью $(n - k)$ -мерного гладкого многообразия

$$\psi_i(\mathbf{x}) = 0, \quad i = \overline{1, k}. \quad (1.14)$$

Далее записываем уравнение абсолютно устойчивой относительно состояний (1.14) в пространстве координат $\psi_i(\mathbf{x}), i = \overline{1, k}$, системы

$$T\dot{\psi}(\mathbf{x}) + \psi(\mathbf{x}) = 0, \quad (1.15)$$

где $\psi(\mathbf{x}) = [\psi_1(\mathbf{x}) \dots \psi_k(\mathbf{x})]^T$, $T > 0$.

Заметим, что при $T > 0$ система (1.15) устойчива относительно начала координат $\psi(t) \rightarrow 0$ при $t \rightarrow \infty$.

Раскрываем соотношение (1.15), получаем

$$\begin{aligned} T \frac{d\psi(\mathbf{x})}{dt} + \psi(\mathbf{x}) &= T \frac{\partial\psi(\mathbf{x})}{\partial\mathbf{x}} \frac{d\mathbf{x}}{dt} + \psi(\mathbf{x}), \\ T \frac{\partial\psi(\mathbf{x})}{\partial\mathbf{x}} \frac{d\mathbf{x}}{dt} + \psi(\mathbf{x}) &= T \frac{\partial\psi(\mathbf{x})}{\partial\mathbf{x}} \mathbf{f}(\mathbf{x}, \mathbf{u}) + \psi(\mathbf{x}), \\ T \frac{\partial\psi(\mathbf{x})}{\partial\mathbf{x}} \mathbf{f}(\mathbf{x}, \mathbf{u}) + \psi(\mathbf{x}) &= 0. \end{aligned} \quad (1.16)$$

Если разрешить систему алгебраических уравнений (1.16) относительно вектора \mathbf{u} , то получим соотношение

$$\mathbf{u} = \mathbf{h}(\mathbf{x}, \psi(\mathbf{x})), \quad (1.17)$$

которое является решением задачи синтеза методом АКАР.

Укажем на существенные недостатки метода АКАР. Во-первых, при решении системы (1.16) вектор управления \mathbf{u} не должен иметь ограничений. Во-вторых, нет гарантии, что система (1.16) разрешима относительно вектора управления \mathbf{u} . В-третьих, решение системы обеспечивает движение, стремящееся на бесконечности к многообразию (1.14), а не удовлетворение уравнениям (1.14), а известно, что движение в окрестности оптимальной траектории часто сильно, с точки зрения критерия оптимальности, отличается от движения по оптимальной траектории. В-четвертых, построение многообразия (1.14) для сложных функционалов (1.4) является трудной задачей, сравнимой по трудности с интуитивным построением самого решения, функции (1.17). Следовательно, метод АКАР использует, как это часто бывает, замену одной проблемы, поиска функции управления (1.5), другой проблемой, построения многообразия (1.14).

Другие аналитические методы синтеза, в основном используют особенности математической модели объекта управления (1.1) и вида функционала (1.4) [33-36]. Часто такие решения приводят к неосуществимым реально системам управления, например в работе [35] для движения робота по траектории система управления использует информацию о кривизне траектории в каждой точке.

Существенным прорывом в решении задачи синтеза управления следует считать появление машинных методов символьной регрессии. Это методы,

которые позволяют с помощью вычислительной машины находить структуру и параметры многомерной функции, описывающей зависимость управления от координат пространства состояний. Отправной точкой для создания таких методов явился метод генетического программирования, разработанный Дж. Козой [53,54]. Метод не предназначался для решения задачи синтеза управления, а разрабатывался для решения задачи автоматического написания программ. Любое математическое выражение вида (1.5) легко реализуется программно, поэтому метод генетического программирования следует считать одним из численных методов для решения задачи синтеза управления.

Численный метод решения задачи синтеза управления выбирает из множества возможных решений математическое выражение (1.5), подставляет его в правые части системы (1.1), интегрирует систему, проверяет выполнение терминальных условий (1.3) и вычисляет значение функционала (1.4). При реализации численного метода возникает ряд вопросов. Во-первых, неясно как вычислять значение функционала для множества начальных значений. Во-вторых, как оценить выполнение терминальных условий, если они выполнены не совсем точно, а значение функционала при этом оказалось удовлетворительным.

Указанные проблемы численного синтеза управления требуют изменения постановки задачи (1.1) - (1.5).

1.2 Многокритериальная задача численного синтеза управления

Задана математическая модель объекта управления (1.1)

Задано множество точек начальных значений

$$X_0 = \{ \mathbf{x}^{0,1}, \dots, \mathbf{x}^{0,M} \}, \quad (1.18)$$

Заданы функционалы качества

$$J_1 = \sum_{i=1}^M \left(\int_0^{t_f} f_0(\mathbf{x}, \mathbf{u}) dt \right)_{\mathbf{x}^{0,i}} \rightarrow \min \quad (1.19)$$

$$J_2 = \sum_{i=1}^M \left(\left\| \varphi(\mathbf{x}(t_f)) \right\| \right)_{\mathbf{x}^{0,i}} \rightarrow \min, \quad (1.20)$$

где нижний индекс $(A)_{\mathbf{x}^{0,i}}$ указывает на вычисление выражения A для начальных условий $\mathbf{x}^{0,i}$, $\|\mathbf{a}\|$ - любая норма вектора \mathbf{a} , $\varphi(\mathbf{x}(t_f)) = [\varphi_1(\mathbf{x}(t_f)) \dots \varphi_l(\mathbf{x}(t_f))]^T$.

Необходимо найти управление в виде

$$\mathbf{u} = \mathbf{g}(\mathbf{x}, \mathbf{q}), \quad (1.21)$$

где \mathbf{q} - вектор искомых параметров, $\mathbf{q} \in \mathbb{R}^p$, $\mathbf{q} = [q_1 \dots q_p]^T$, $\mathbf{g}(\mathbf{x}, \mathbf{q})$ - искомая функция, $\mathbf{g}(\mathbf{x}, \mathbf{q}) : \mathbb{R}^n \times \mathbb{R}^p \rightarrow \mathbb{R}^m$, $\mathbf{g}(\mathbf{x}, \mathbf{q}) = [g_1(\mathbf{x}, \mathbf{q}) \dots g_m(\mathbf{x}, \mathbf{q})]^T$

Поскольку задача (1.1), (1.18) - (1.21) является многокритериальной, то ее решением является множество Парето

$$P = \left\{ \tilde{\mathbf{g}}^1(\mathbf{x}, \tilde{\mathbf{q}}), \dots, \tilde{\mathbf{g}}^D(\mathbf{x}, \tilde{\mathbf{q}}) \right\}, \quad (1.22)$$

Для элементов множества Парето P выполняются условия: $\forall \mathbf{g}(\mathbf{x}, \mathbf{q}) \notin P$
 $\exists \tilde{\mathbf{g}}^j(\mathbf{x}, \tilde{\mathbf{q}}) \in P$, для которого выполняется неравенство

$$\mathbf{J}(\tilde{\mathbf{g}}^j(\mathbf{x}, \tilde{\mathbf{q}})) \leq \mathbf{J}(\mathbf{g}(\mathbf{x}, \mathbf{q})), \quad (1.23)$$

где $\mathbf{J}(\mathbf{a}(\mathbf{x}, \mathbf{q})) = [J_1(\mathbf{a}(\mathbf{x}, \mathbf{q})) \ J_2(\mathbf{a}(\mathbf{x}, \mathbf{q}))]^T$, $J_i(\mathbf{a}(\mathbf{x}, \mathbf{q}))$ - значение функционала J_i , вычисленное для функции $\mathbf{a}(\mathbf{x}, \mathbf{q}) = \mathbf{g}(\mathbf{x}, \mathbf{q}) \vee \tilde{\mathbf{g}}^j(\mathbf{x}, \tilde{\mathbf{q}})$, $i = \overline{1,2}$.

Неравенство векторов (1.23) выполняется в том случае, когда все компоненты одного вектора не больше компонент другого вектора, и существует одна компонента, значение которой строго меньше значения этой же компоненты другого вектора

$$J_i(\tilde{\mathbf{g}}^j(\mathbf{x}, \tilde{\mathbf{q}})) \leq J_i(\mathbf{g}(\mathbf{x}, \mathbf{q})), \quad i = \overline{1,2}. \quad J_k(\tilde{\mathbf{g}}^j(\mathbf{x}, \tilde{\mathbf{q}})) < J_k(\mathbf{g}(\mathbf{x}, \mathbf{q})), \quad k \in \{1,2\}. \quad (1.24)$$

В 2006 г. для решения задачи численного многокритериального синтеза системы управления (1.1), (1.18) - (1.21) профессором А.И. Дивеевым из Вычислительного центра им. А.А. Дородницына Российской академии наук был разработан метод сетевого оператора [7-10,15,16,49-52], который позволяет представлять математическое выражение в виде целочисленной матрицы.

Эффективность метода сетевого оператора определяется быстрым вычислением математического выражения по матрице сетевого оператора и использованием при поиске принципа малых вариаций базисного решения. Рассмотрим метод сетевого оператора подробнее.

1.3 Метод сетевого оператора

В методе сетевого оператора математическое выражение описывается с помощью ориентированного графа. Структура графа указывает на порядок выполнения операций и на аргументы любой функции при вычислении математического выражения.

Сетевой оператор строят на следующих конструктивных множествах:

- множество переменных

$$X = (x_1, \dots, x_N), \quad x_i \in \mathbb{R}^1, \quad i = \overline{1, N}; \quad (1.25)$$

- множество параметров

$$Q = (q_1, \dots, q_P), \quad q_i \in \mathbb{R}^1, \quad i = \overline{1, P}; \quad (1.26)$$

- множество унарных операций

$$O_1 = (\rho_1(z) = z, \rho_2(z), \dots, \rho_W(z)); \quad (1.27)$$

- множество бинарных операций

$$O_2 = (\chi_0(z', z''), \dots, \chi_{V-1}(z', z'')). \quad (1.28)$$

Среди унарных операций обязательно должна присутствовать тождественная операция

$$\rho_1(z) = z. \quad (1.29)$$

Бинарные операции должны быть коммутативны

$$\chi_i(z', z'') = \chi_i(z'', z'), \quad i = \overline{0, V-1}, \quad (1.30)$$

ассоциативны

$$\chi_i(z', \chi_i(z'', z''')) = \chi_i(\chi_i(z', z''), z'''), \quad i = \overline{0, V-1}, \quad (1.31)$$

и иметь единичный элемент

$$\forall \chi_i(z', z'') \in O_2 \quad \exists e_i \Rightarrow \chi_i(e_i, z) = z, \quad i = \overline{0, V-1}. \quad (1.32)$$

Нумерация элементов множества O_2 бинарных операций в отличие от элементов множества O_1 унарных операций начинается с нуля. Это вызвано частым применением в качестве бинарных операций двух основных операций сложения и умножения. Единичными элементами для этих операций являются ноль и единица. Для простоты запоминания номера основных операций совпадают со значениями их единичных элементов.

Номера элементов множества O_1 унарных операций не могут начинаться с нуля. При описании структуры данных сетевого оператора в форме матрицы ноль в позиции для унарной операции означает ее отсутствие.

Сетевой оператор - это ориентированный граф, обладающий следующими свойствами:

- а) в графе отсутствуют циклы;
- б) к любому узлу, который не является источником, имеется хотя бы один путь от узла-источника;
- в) от любого узла, который не является стоком, имеется хотя бы один путь до узла-стока;
- г) каждому узлу-источнику соответствует элемент из множества переменных X или из множества параметров Q ;
- д) каждому узлу соответствует бинарная операция из множества O_2 бинарных операций;
- е) каждой дуге графа соответствует унарная операция из множества O_1 унарных операций.

Любой ориентированный граф без циклов с указанием параметров и переменных в узлах-источниках, номеров бинарных операций в остальных узлах и номеров унарных операций на дугах соответствует некоторому математическому выражению.

Правила записи элементов математического выражения по графу сетевого оператора приведены на рисунок 1.1.

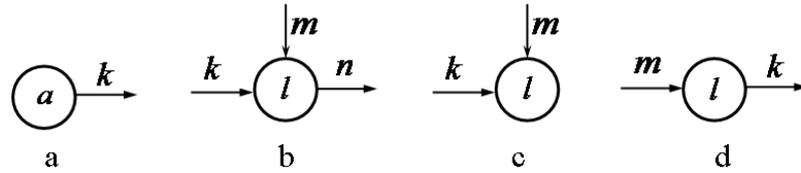


Рисунок 1.1 Элементы графа сетевого оператора: a - $\rho_k(a)$, $a \in X \cup Q$;
 b - $\rho_n(\chi_l(\rho_k(z'), \rho_m(z'')))$; c - $\chi_l(\rho_k(z'), \rho_m(z''))$; d - $\rho_k(\chi_l(\rho_m(z)), e_l) = \rho_k(\rho_m(z))$

Рассмотрим в качестве примера процесс построения сетевого оператора для следующего математического выражения

$$y = x_1 + \sin(x_1) + q_1 x_2 e^{-q_2 x_2}.$$

Зададим конструктивные множества

$$X = (x_1, x_2), Q = (q_1, q_2), O_1 = (\rho_1(z) = z, \rho_2(z) = -z, \rho_3(z) = e^z, \rho_4(z) = \sin(z)),$$

$$O_2 = (\chi_0(z_1, z_2) = z_1 + z_2, \chi_1(z_1, z_2) = z_1 \cdot z_2).$$

Сетевой оператор рассмотренного математического выражения представлен на рисунок 1.2.

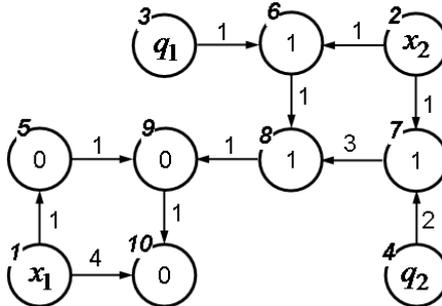


Рисунок 1.2 Сетевой оператор примера математического выражения

На рисунок 1.2 в верхних частях вершин указаны их номера, установленные в соответствие с требованиями топологической сортировки, когда номер узла, откуда дуга выходит, всегда меньше номера узла, куда дуга входит.

Для представления сетевого оператора в памяти компьютера используют матрицу сетевого оператора. Матрица сетевого оператора имеет структуру матрицы смежности графа сетевого оператора. Вместо недиагональных единиц в матрице сетевого оператора указывают номера унарных операций, с которой связана дуга, соответствующая единице в матрице смежности. На диагонали матрицы сетевого оператора указывают номер бинарной операции, которая

связана с узлом графа, номер которого соответствует номеру строки матрицы, если данная строка матрицы не определяет узел-источник. При топологической сортировке узлов графа сетевого оператора матрица сетевого оператора имеет верхнюю треугольную форму.

Для рассматриваемого примера матрица сетевого оператора имеет следующий вид:

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

Нулевые столбцы матрицы сетевого оператора соответствуют узлам-источникам сетевого оператора. Нулевые строки матрицы сетевого оператора соответствуют узлам-стокам сетевого оператора.

Для вычисления математического выражения по матрице сетевого оператора используем соотношение

$$z_j^{(i)} = \begin{cases} \chi_{\Psi_{j,j}} \left(\rho_{\Psi_{i,j}} \left(z_i^{(i-1)} \right), z_j^{(i-1)} \right), & \text{если } \Psi_{i,j} \neq 0, \quad i = \overline{1, L-1}, \quad j = \overline{i+1, L}, \\ z_j^{(i-1)} & \text{иначе} \end{cases}, \quad (1.33)$$

где L - число строк матрицы сетевого оператора, $z_j^{(i)}$ - компонента j вектора узлов

$$\mathbf{z} = [z_1 \ \dots \ z_L]^T \quad (1.34)$$

на итерации i .

Перед вычислением математического выражения по формуле (1.33) необходимо задать начальное значение вектора узлов

$$\mathbf{z}^{(0)} = [z_1^{(0)} \ \dots \ z_L^{(0)}]^T.$$

Каждая компонента вектора узлов соответствует узлу сетевого оператора. Начальное значение компоненты вектора узлов для узлов-источников равно

значению связанного с этим узлом элемента множества переменных или параметров

$$z_j^{(0)} = a \in X \cup Q, \text{ если } \psi_{i,j} = 0, i = \overline{1, L}. \quad (1.34)$$

Начальные значения остальных компонент вектора узлов равны единичным элементам связанных с этими узлами бинарных операций.

$$z_j^{(0)} = \psi_{j,j}, \text{ если } \exists \psi_{i,j} \neq 0, 1 \leq j \leq L. \quad (1.35)$$

Количество итераций при вычислении математического выражения по матрице сетевого оператора на единицу меньше количества строк в матрице сетевого оператора.

Для поиска оптимального решения в виде матрицы сетевого оператора используют метод малых вариаций базисного решения. Чаще всего при решении задач методом сетевого оператора применяют следующие малые вариации:

- 0 - замена унарной операции;
- 1 - замена бинарной операции;
- 2 - добавление унарной операции;
- 3 – удаление унарной операции.

Для описания малых вариации используем вектор из четырех компонент

$$\mathbf{w} = [w_1 \ w_2 \ w_3 \ w_4]^T, \quad (1.36)$$

где w_1 - номер малой вариации, w_2 - номер строки, w_3 - номер столбца, w_4 - номер унарной или бинарной операции.

Например, для малой вариации $\mathbf{w}^1 = [0 \ 2 \ 7 \ 3]^T$ для рассмотренного примера, получаем вариацию 0, изменение значения унарной операции в строке 2 в столбце 7 на операцию 3. В результате получаем

$$\mathbf{w}^1 \circ \Psi = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 4 \\ 0 & 0 & 0 & 0 & 0 & 1 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Полученная новая матрица сетевого оператора соответствует математическому выражению

$$y = q_1 x_2 e^{-q_2 e^{x_2}} + x_1 + \sin x_1.$$

При решении задачи синтеза системы управления методом сетевого оператора задают одно базисное решение в виде матрицы сетевого оператора

$$\Psi^0 = [\psi_{i,j}^0], \quad i, j = \overline{1, L}, \quad (1.37)$$

и множество наборов векторов вариаций

$$G = (W_0, W_1, \dots, W_H), \quad (1.38)$$

где

$$W_i = (\mathbf{w}^{i,1}, \dots, \mathbf{w}^{i,d}). \quad (1.39)$$

Базисное решение определяется в виде математического выражения разработчиком. При выборе базисного решения разработчик исследует особенности задачи и свойства объекта управления. Для получения базисного решения целесообразно использовать приближенные и аналитические методы. Часто для получения базисного решения снижают требования к одному из функционалов (1.19), (1.20), например, к терминальному функционалу. Затем в процессе поиска базисное решение уточняют по результатам математического выражения, отобранного из найденного множества Парето. Данное решение считают базисным и повторяют поиск.

Каждое возможное решение определяется малыми вариациями базисного решения (1.37)

$$\Psi^i = \mathbf{w}^{i,d} \circ \dots \circ \mathbf{w}^{i,1} \circ \Psi^0. \quad (1.40)$$

Во множестве возможных решений (1.38), как правило, задают набор из тождественных вариаций. Данный набор не меняет базисного решения, но участвует в поиске. Во множестве возможных решений (1.38) набор тождественных векторов вариаций определен в виде нулевого элемента W_0 .

Вектор тождественных вариаций определяют всеми нулевыми компонентами

$$\mathbf{w}^{0,j} = [0 \ 0 \ 0 \ 0]^T, \quad \Psi = [0 \ 0 \ 0 \ 0]^T \circ \Psi, \quad j \in \{1, \dots, d\}. \quad (1.41)$$

При поиске решения методом сетевого оператора используют генетический алгоритм многокритериальной оптимизации. Операцию скрещивания при поиске структуры выполняют на отобранных наборах векторов вариаций. Скрещивание выполняют классическим для генетического алгоритма способом, обменом элементов в структурах после случайно определенной точки скрещивания. В результате скрещивания получают два новых набора векторов вариаций

$$W_{H+1} = (\mathbf{w}^{i_1,1}, \dots, \mathbf{w}^{i_1,k_s}, \mathbf{w}^{i_2,1}, \dots, \mathbf{w}^{i_2,d}), \quad (1.42)$$

$$W_{H+2} = (\mathbf{w}^{i_2,1}, \dots, \mathbf{w}^{i_2,k_s}, \mathbf{w}^{i_1,1}, \dots, \mathbf{w}^{i_1,d}), \quad (1.43)$$

где $W_{i_1} = (\mathbf{w}^{i_1,1}, \dots, \mathbf{w}^{i_1,d})$, $W_{i_2} = (\mathbf{w}^{i_2,1}, \dots, \mathbf{w}^{i_2,d})$ - выбранные для скрещивания элементы множества (1.38) возможных решений, $i_1, i_2 \in \{0, \dots, H\}$, k_s - случайно определенная точка скрещивания, $1 \leq k_s \leq d$.

Для оценки решения используют ранг Парето – это величина, которая указывает на количества элементов во множестве возможных решение (1.38), которые по соотношению Парето лучше, чем данное решение. Согласно рангу Парето лучшими во множестве возможных решений являются решения с нулевым рангом, поэтому множество Парето строится из этих решений.

При оценке новых возможных решений (1.42), (1.43) определяют номер решения с наибольшим значением ранга Парето. Затем вычисляют значение ранга Парето для одного нового решения. Если ранг Парето нового решения оказывается меньше, чем наибольший ранг, то возможное решение с наибольшим рангом заменяют новым полученным возможным решением.

Следует отметить, что в соответствии с эволюционным поиском множество возможных решений изменяется, поэтому ранги Парето для каждого решения и множество Парето в процессе поиска также меняются.

В процессе поиска решения одновременно ищут и оптимальное значение вектора параметров также с помощью эволюционного алгоритма поиска. После выполнения нескольких эволюционных циклов отбирают множество Парето, находят в нем наиболее подходящее решение задачи и меняют базисное решение на новое найденное решение. Если в процессе поиска не было найдено решение лучше базисного, то оставляют базисное решение.

1.4 Выводы к главе 1

1. Приведена формальная постановка задачи синтеза управления. Показано, что особенностью задачи синтеза является наличие области начальных условий, обязательное выполнение терминальных условий и свойства самого решения, которое должно представлять собой математическое выражение, описывающее функциональную зависимость управления от координат пространства состояний.

2. Приведен обзор аналитических методов для решения задачи синтеза управления. Показано, что основным результатом аналитического решения задачи синтеза управления является метод аналитического конструирования оптимальных регуляторов, который для линейной модели объекта управления и квадратичного функционала находит решение в форме линейной функции от координат пространства состояний. Приведено краткое описание метода аналитического конструирования агрегированных регуляторов, который также относится к аналитическим методам.

3. Приведена постановка задачи численного синтеза системы управления в форме многокритериальной задачи структурно-параметрической оптимизации. В задачи численного синтеза область начальных значений заменена множеством точек, терминальные условия определены в виде дополнительного функционала, значения функционалов вычисляются в виде сумм функционалов, подсчитанных для каждого начального значения.

4. Приведено краткое описание метода сетевого оператора. Показано, что преимуществом метода сетевого оператора, в частности перед методом генетического программирования, являются эффективное вычисление математического выражения по целочисленной матрице оператора и использование при поиске принципа малых вариаций базисного решения.

2. МЕТОД ВАРИАЦИОННОГО ГЕНЕТИЧЕСКОГО ПРОГРАММИРОВАНИЯ

Методы, на основе идей которых был создан метод вариационного генетического программирования, представлены на рисунк 2.1

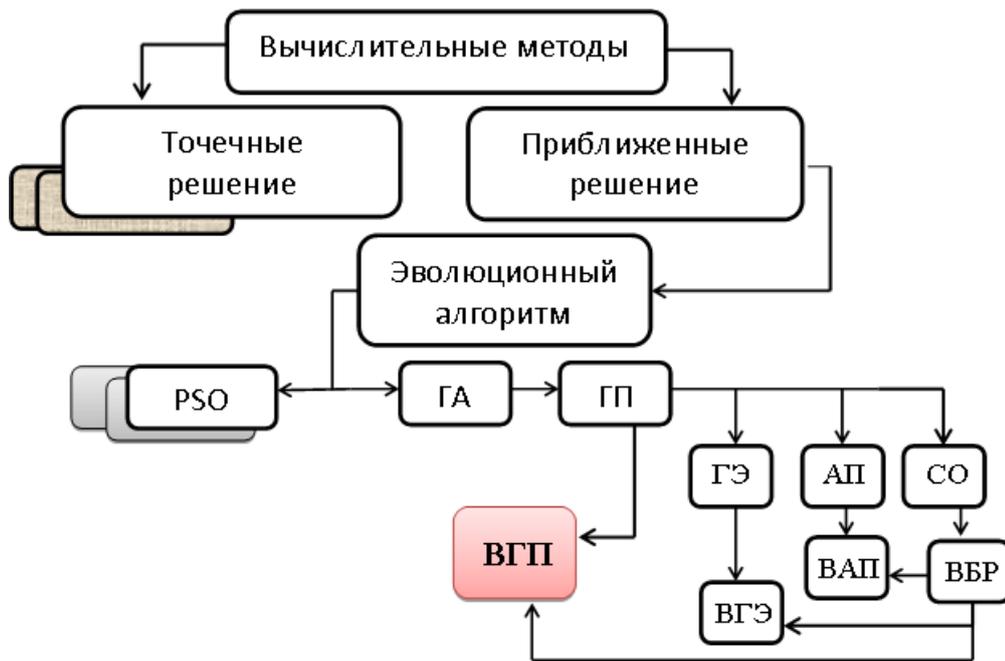


Рисунок 2.1 Источники создания метода вариационного генетического программирования

На Рисунок 2.1 используются следующие обозначения: PSO – метод роя частиц, ГА – генетический алгоритм, ГП – метод генетического программирования, ГЭ – метод грамматической эволюции, АП – метод аналитического программирования, СО – метод сетевого оператора, ВГП – метод вариационного генетического программирования, ВАП – метод вариационного аналитического программирования, ВБР – метод вариаций базисного решения.

Как видно из рисунка метод вариационного генетического программирования [11,14] относится к эволюционным методам поиска решения. Основными источниками для создания метода вариационного генетического программирования являются метод генетического программирования и метод сетевого оператора. Метод сетевого оператора был описан ранее в разделе 1. Рассмотрим подробнее метод генетического программирования

2.1. Метод генетического программирования

Метод генетического программирования [53, 54] использует эволюционный генетический алгоритм для поиска оптимальной структуры, описываемой строкой символов. При поиске математического выражения символы в строке обозначают функции с различным количеством аргументов, в том числе переменные и параметры, которые считаем функциями с нулевым количеством аргументов или нульместными функциями. Правила записи математического выражения в форме набора строк соответствуют префиксной символьной бесскобочной польской записи, используемой при создании трансляторов языков программирования высокого уровня. Генетическое программирование создавалось для решения задачи автоматического написания программ.

Каждый элемент записи в генетическом программировании определенный элемент математического выражения. По правилам в префиксной записи код функции предшествует кодам ее аргументов. В графической интерпретации код записи математического выражения соответствует дереву. Префиксная запись кодирует обход дерева сверху вниз слева направо.

Рассмотрим в качестве примера математическое выражение

$$y_1 = e^{-q_1 x_1} \sin(\cos(q_2 x_2)). \quad (2.1)$$

Символы, которые кодируют элементы данного математического выражения, приведены в таблице 1. Во втором столбце таблицы указано количество необходимых аргументов для функции. Переменные и параметры имеют ноль аргументов или соответствуют нульместным функциям.

Согласно правилам префиксной записи строка символов, кодирующая данное математическое выражение, имеет вид

$$S(y_1) = mefmcaghmdb. \quad (2.2)$$

Граф префиксного кода записи математического выражения приведен на рисунок 2.2.

Для вычисления математического выражения по графу выполняем обход дерева сверху вниз, слева направо.

Таблица 2.1 Коды элементов математического выражения

Операция	Число аргументов	Обозначение
x_1	0	a
x_2	0	b
q_1	0	c
q_2	0	d
e^z	1	e
$-z$	1	f
$\sin(z)$	1	g
$\cos(z)$	1	h
$z_1 + z_2$	2	s
$z_1 z_2$	2	m

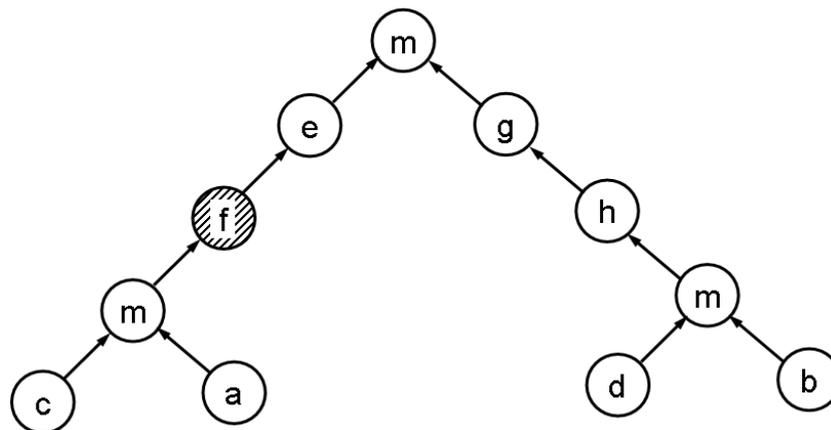


Рисунок 2.2 Граф вычислений математического выражения (2.1)

Согласно таблице 2.1 получаем

$$m \rightarrow z_1 z_2, \quad y_1 = z_1 z_2;$$

$$e \rightarrow e^z, \quad y_1 = e^z z_1;$$

$$f \rightarrow -z, \quad y_1 = e^{-z} z_1;$$

$$m \rightarrow z_1 z_2, \quad y_1 = e^{-z_1 z_2} z;$$

$$c \rightarrow q_1, \quad y_1 = e^{-q_1 z_2} z;$$

$$a \rightarrow x_1, \quad y_1 = e^{-q_1 x_1} z;$$

$$g \rightarrow \sin(z), y_1 = e^{-q_1 x_1} \sin(z);$$

$$h \rightarrow \cos(z), y_1 = e^{-q_1 x_1} \sin(\cos(z));$$

$$m \rightarrow z_1 z_2, y_1 = e^{-q_1 x_1} \sin(\cos(z_1 z_2));$$

$$d \rightarrow q_2, y_1 = e^{-q_1 x_1} \sin(\cos(q_2 z));$$

$$b \rightarrow x_2, y_1 = e^{-q_1 x_1} \sin(\cos(q_2 x_2)).$$

В приведенной последовательности вычислений начало каждой строки соответствует символу из кода префиксной записи (2.2), поэтому наличие графа вычислений для интерпретации математического выражения по коду записи не требуется. Достаточно иметь строку символов префиксной записи (2.2).

При поиске оптимального решения в виде кода записи математического выражения в методе генетического программирования используется эволюционный генетический алгоритм. Особенностью выполнения данного алгоритма является специальная операция скрещивания, которая для символьных строк выполняется в точках, определяемых символами, описывающими функциями с одинаковым числом аргументов. Операция выполняется с помощью обмена подстрок, соответствующих поддеревьям, корнями которых являются точки скрещивания.

Рассмотрим операцию скрещивания подробнее. Пусть задано еще одно математическое выражение

$$y_2 = \sin(q_2 + x_2) + \cos(q_1 x_1). \quad (2.3)$$

Согласно таблице 2.1 код записи выражения (2.3) имеет вид

$$S(y_2) = sgmdbhmca. \quad (2.4)$$

Вычислительный граф символьной записи (2.4) математического выражения (2.3) приведен на рис 2.3.

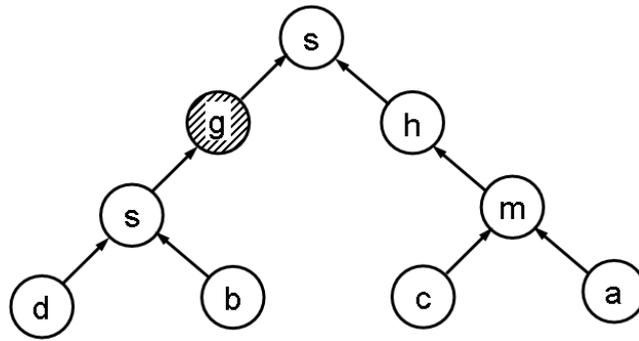


Рисунок 2.3 Граф вычислений математического выражения (2.3)

Пусть в обеих строках символов выбраны точки скрещивания, которые соответствуют функциям с одним аргументом. Точки скрещивания на рисунках 2.2 и 2.3 отмечены заштрихованными вершинами. Обменим поддеревья, которые начинаются с этих вершин. Получаем новые графы вычислений. Графы указаны на рисунках 2.4 и 2.5.

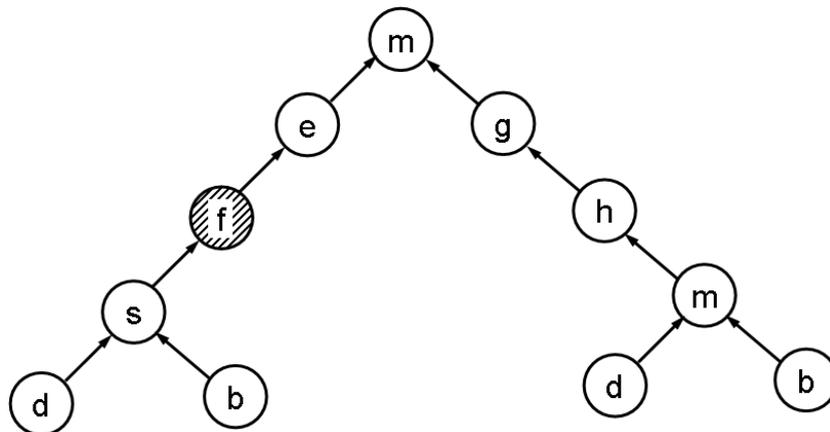


Рисунок 2.4 Граф вычислений первого нового математического выражения

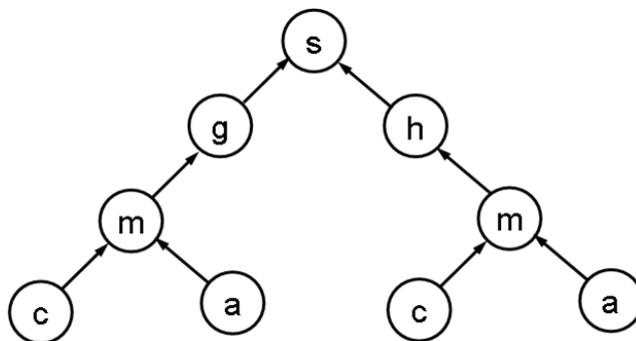


Рисунок 2.5 Граф вычислений второго нового математического выражения

Полученные математические выражения имеют следующие коды записи:

$$S(y_3) = mefsdbghmdb, \quad (2.5)$$

$$S(y_4) = sgmcahmca. \quad (2.6)$$

Полученные новые коды записей соответствуют новым математическим выражениям

$$y_3 = e^{-(q_2 + x_2)} \sin(\cos(q_2 x_2)), \quad (2.7)$$

$$y_4 = \sin(q_1 x_1) + \cos(q_1 x_1). \quad (2.8)$$

При выполнении операции скрещивания необходимо находить подстроки символов, соответствующие обмениваемым поддеревьям. Это требует использования специального алгоритма, который бы определял правильность завершения вычислений по поддереву. Другим недостатком операции скрещивания в генетическом программировании является различные длины обмениваемых подстрок, что приводит к получению кодов записей различной длины. Практика использования генетического программирования показывает, что многократное выполнение операций скрещивания приводит к появлению длинных записей, так как в таких записях наиболее вероятно нахождение точек скрещивания соответствующих указанным выше требованиям. Для сокращения длин записей в генетическом программировании используют ограниченное число символов. Если при скрещивании получается записей с количеством символов, превышающем заданную величину, то скрещивание не выполняют. Такое дополнительное ограничение уменьшает число скрещиваний во множестве возможных решений и сокращает вероятность эволюции множества.

Генетическую операцию мутации в генетическом программировании либо не выполняют, либо выполняют с помощью замены случайно выбранного в коде записи символа на случайно отобранный из таблицы символ, соответствующей функции с тем же количеством аргументов, что заменяемый символ.

Метод генетического программирования позволяет решать задачу синтеза системы управления [38, 39, 45, 46, 54]. Для решения данной задачи требуется

большое множество возможных решений и большое количество итераций по эволюции данного множества.

Метод генетического программирования при решении задачи синтеза системы управления обладает следующими недостатками:

- вычисление математического выражения требует анализа всех символов кода с целью их перевода в соответствующие элементы математического выражения;
- выполнение операции скрещивания требует нахождения в строках символов обмениваемых подстрок, которые соответствуют поддеревьям вычислений;
- выполнение операции скрещивания требует выполнения алгоритма обмена подстрок разной длины в кодах записей математических выражений;
- длина кодов записей математических выражений не одинакова и в процессе поиска многократного скрещивания имеет тенденцию к увеличению;
- число вхождений параметров и переменных в математическое выражение, описываемое кодом из строк символов, равно числу символов, соответствующих этим параметрам и переменным.

Недостатки метода генетического программирования, используемого для решения задачи синтеза системы управления, определены не очень удобной для алгоритмического поиска символьной структурой данных. Основная проблема, которая возникает при поиске решения задачи синтеза методом генетического программирования – это длительное время счета, которое определено большим множеством возможных решений и наличием интерпретатора или лексического анализатора, обеспечивающих перевод символьных данных в элементы математического выражения.

2.2 Коды символов метода генетического программирования

Рассмотрим усовершенствованный вид кодирования символов в методе генетического программирования.

Для построения дерева вычислений и выполнения самого вычисления по коду записи математического выражения интерпретатору необходимо из кода символа получить наименование функции и количество используемых ею аргументов. Следовательно, каждый символ кода записи математического выражения должен содержать информацию о функции и количестве аргументов.

Для кодирования элементов математического выражения введём множество упорядоченных наборов функций с определённым количеством аргументов

$$F = \{F_0, \dots, F_n\}, \quad (2.9)$$

где

$$F_i = (f_{i,1}(z_1, \dots, z_i), \dots, f_{i,m_i}(z_1, \dots, z_i)), \quad i = \overline{0, n}, \quad (2.10)$$

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

В качестве кода функции используем целочисленный вектор из двух элементов

$$\mathbf{s} = [s_1 \ s_2]^T, \quad (2.11)$$

где s_1 - количество аргументов функции, s_2 - номер функции в множестве F_{s_1} .

Согласно принятым обозначениям для функции $f_j(z_1, \dots, z_i)$ получаем

$$\mathbf{s} = [i \ j]^T. \quad (2.12)$$

Математическое выражение представляет собой упорядоченное множество кодов функций

$$S = (\mathbf{s}^1, \dots, \mathbf{s}^K), \quad (2.13)$$

где $\mathbf{s}^i = [s_1^i \ s_2^i]^T$, $i = \overline{1, K}$.

Рассмотрим пример. Пусть имеем математическое выражение (2.1)

Для данного выражения имеем следующие множества функций

$$F_0 = (x_1, x_2, q_1, q_2),$$

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

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

Запишем функции в кодах

$$\bar{F}_0 = \left(\begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 2 \end{bmatrix}, \begin{bmatrix} 0 \\ 3 \end{bmatrix}, \begin{bmatrix} 0 \\ 4 \end{bmatrix} \right), \quad \bar{F}_1 = \left(\begin{bmatrix} 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \begin{bmatrix} 1 \\ 3 \end{bmatrix}, \begin{bmatrix} 1 \\ 4 \end{bmatrix} \right), \quad \bar{F}_2 = \left(\begin{bmatrix} 2 \\ 1 \end{bmatrix}, \begin{bmatrix} 2 \\ 2 \end{bmatrix} \right),$$

где \bar{F}_i - множество кодов функций множества F_i .

Запись математического выражения (2.1) в рассматриваемом коде имеет вид

$$S = \left(\begin{bmatrix} 2 \\ 2 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \begin{bmatrix} 2 \\ 2 \end{bmatrix}, \begin{bmatrix} 0 \\ 3 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 3 \end{bmatrix}, \begin{bmatrix} 1 \\ 4 \end{bmatrix}, \begin{bmatrix} 2 \\ 2 \end{bmatrix}, \begin{bmatrix} 0 \\ 4 \end{bmatrix}, \begin{bmatrix} 0 \\ 2 \end{bmatrix} \right). \quad (2.14)$$

На рисунок 2.6 приведен граф вычислений для математического выражения (2.1)

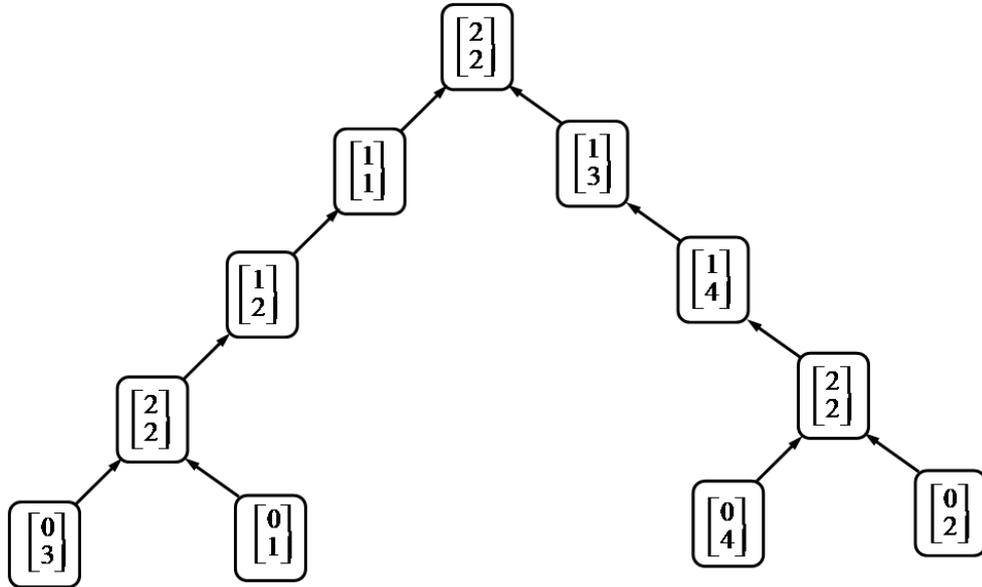


Рисунок 2.6 Граф вычислений в цифровой кодировке математического выражения (2.1)

При составлении кода математического выражения используем правила:

- а) код первого аргумента функции следует непосредственно сразу после кода функции;
- б) коды второго и следующих аргументов функции следуют после кода функции с нулевым количеством аргументов;
- в) каждый код аргумента относится к ближайшей слева функции с недостающим количеством аргументов.

Из установленных правил следуют определенные свойства кода записи математического выражения.

Для определения правильности записи кода математических выражений используем понятие индекса символа записи. Индекс $T(j)$ символа s^j кода (2.13) математического выражения указывает на минимальное количество недостающих справа кодов символов

$$T(j) = 1 - j + \sum_{i=1}^j s_1^i, \quad (2.15)$$

где

$$T(j) > 0, \quad j = \overline{1, K-1}, \quad (2.16)$$

$$T(K) = 0, \quad (2.17)$$

В качестве примера подсчитаем коды символов выражения (2.14). Код первого символа $s^1 = [2 \ 2]^T$ равен

$$T(1) = 1 - 1 + 2 = 2.$$

Действительно код первого символа – это код функции с двумя аргументами, поэтому после этого символа необходимо наличие не менее двух символов, кодов аргументов.

Подсчитываем индексы остальных символов, получаем

$$T(2) = 1 - 2 + 2 + 1 = 2,$$

$$T(3) = 1 - 3 + 2 + 1 + 1 = 2,$$

$$T(4) = 1 - 4 + 2 + 1 + 1 + 2 = 3,$$

$$T(5) = 1 - 5 + 2 + 1 + 1 + 2 + 0 = 2,$$

$$T(6) = 1 - 6 + 2 + 1 + 1 + 2 + 0 + 0 = 1,$$

$$T(7) = 1 - 7 + 2 + 1 + 1 + 2 + 0 + 0 + 1 = 1,$$

$$T(8) = 1 - 8 + 2 + 1 + 1 + 2 + 0 + 0 + 1 + 1 = 1,$$

$$T(9) = 1 - 9 + 2 + 1 + 1 + 2 + 0 + 0 + 1 + 1 + 2 = 2,$$

$$T(10) = 1 - 10 + 2 + 1 + 1 + 2 + 0 + 0 + 1 + 1 + 2 + 0 = 1,$$

$$T(11) = 1 - 11 + 2 + 1 + 1 + 2 + 0 + 0 + 1 + 1 + 2 + 0 + 0 = 0.$$

Индекс символа позволяет определить правильность кода записи математического выражения. Для правильной записи должны выполняться условия (2.16), (2.17). Индекс любого символа в записи, кроме последнего, должен быть больше нуля (2.16). Индекс последнего символа должен быть равен нулю (2.17).

Согласно соотношению (2.15) код символа параметра или переменной уменьшают величину индекса на единицу. Код символа, описывающего функцию с одним аргументом, не изменяют величину индекса. Действительно из (2.15) получаем

$$T(j) = T(j-1) - 1 + s_1^j, \quad (2.18)$$

поэтому, если $s_1^j = 1$, то $T(j) = T(j-1)$.

Коды функций с двумя и более аргументами увеличивают величину индекса.

По индексу можно всегда построить правильную запись кодов математического выражения. Пусть в выражении (2.14) код седьмого символа $T(7) = 1 - 7 + 2 + 1 + 1 + 2 + 0 + 0 + 1 = 1$. Это означает, что, если восьмым символом будет символ кода параметра или переменной, т.е. код функции без аргументов, $s_1^8 = 0$, то строка кодов будет завершенной кодированной строкой правильного математического выражения. Введем в запись (2.14) в качестве восьмого символа код переменной $s^8 = [0 \ 2]^T$, получаем

$$S = \left(\left[\begin{array}{c} 2 \\ 2 \end{array} \right], \left[\begin{array}{c} 1 \\ 1 \end{array} \right], \left[\begin{array}{c} 1 \\ 2 \end{array} \right], \left[\begin{array}{c} 2 \\ 2 \end{array} \right], \left[\begin{array}{c} 0 \\ 3 \end{array} \right], \left[\begin{array}{c} 0 \\ 1 \end{array} \right], \left[\begin{array}{c} 1 \\ 3 \end{array} \right], \left[\begin{array}{c} 0 \\ 2 \end{array} \right] \right).$$

Полученная запись соответствует правильному математическому выражению

$$y_1 = e^{-q_1 x_1} \sin(x_2).$$

Индекс (2.15) символа кода записи позволяет определять подстроки выражений, соответствующие поддеревьям вычислений. Нахождение подстрок необходимо для выполнения операции скрещивания в генетическом алгоритме. Для нахождения подстроки необходимо символ, который является начальным в подстроке, считать первым. После этого необходимо подсчитывать индексы символов по формуле (2.15), как индексы кодов отдельного математического выражения до тех пор, пока не встретим символ с индексом равным нулю. Этот символ будет последним символом подстроки.

Рассмотрим пример нахождения подстроки. При выполнении операции скрещивания в приведенном выше примере в коде записи математического выражения (2.1) точкой скрещивания был выбран третий символ « f ». В цифровом коде записи (2.14) этот символ соответствует коду $\mathbf{s}^3 = [1 \ 2]^T$ функции $f_{1,2}(z) = -z$. Определим подстроку кода, считая этот символ первым символом математического выражения. Вычисления выполняем, пока не получим нулевой индекс.

Индекс третьего символа, если считать его первым имеет значение

$$T(3) = 1 - 1 + 1 = 1.$$

Далее вычисляем

$$T(4) = 1 - 2 + 1 + 2 = 2,$$

$$T(5) = 1 - 3 + 1 + 2 + 0 = 1,$$

$$T(6) = 1 - 4 + 1 + 2 + 0 + 0 = 0.$$

Следовательно, в подстроке четыре символа $(\mathbf{s}^3, \mathbf{s}^4, \mathbf{s}^5, \mathbf{s}^6)$

$$\mathbf{S} = \left(\begin{bmatrix} 1 \\ 2 \end{bmatrix}, \begin{bmatrix} 2 \\ 2 \end{bmatrix}, \begin{bmatrix} 0 \\ 3 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right).$$

Подстрока соответствует математическому выражению

$$\tilde{y} = -q_1 x_1$$

или части дерева вычислений на Рисунок 2.5.

В общем случае запись кода подвыражения является подмножеством символов кода записи математического выражения

$$\mathbf{S}(m) = (\mathbf{s}^m, \mathbf{s}^{m+1}, \dots, \mathbf{s}^{m+k}) \subseteq \mathbf{S}. \quad (2.19)$$

Индекс символов подстроки $\mathbf{S}(m)$ вычисляем по формуле

$$T_m(j) = m - j + \sum_{i=m}^j s_1^i, \quad j = m, m+1, \dots, m+k. \quad (2.20)$$

Из формулы (2.20) получаем

$$T_m(m) = s_1^m,$$

$$T_m(m+1) = -1 + s_1^m + s_1^{m+1},$$

$$T_m(m+2) = -2 + s_1^m + s_1^{m+1} + s_1^{m+2} = T_m(m+1) - 1 + s_1^{m+2}.$$

Формуле (2.20) соответствует рекуррентное соотношение

$$T_m(m+j) = T_m(m+j-1) - 1 + s_1^{m+j}. \quad (2.21)$$

Индекс последнего символа подстроки равен нулю

$$T_m(m+k) = 0. \quad (2.22)$$

Соотношение (2.22) позволяет осуществить проверку окончания подстроки.

Операция скрещивания в генетическом программировании осуществляется обменом подстрок. Пусть имеем две строки кодов длиной L_1 и L_2 .

$$S_1 = (s^{1,1}, \dots, s^{1,L_1}), S_2 = (s^{2,1}, \dots, s^{2,L_2}). \quad (2.23)$$

Находим в строках подстроки

$$S_1(m_1) = (s^{1,m_1}, \dots, s^{1,m_1+k_1}), S_2(m_2) = (s^{2,m_2}, \dots, s^{2,m_2+k_2}). \quad (2.24)$$

где

$$s^{1,j_1} = \begin{bmatrix} s_1^{1,j_1} & s_2^{1,j_1} \end{bmatrix}^T, \quad j_1 = m_1, m_1 + 1, \dots, m_1 + k_1,$$

$$s^{2,j_2} = \begin{bmatrix} s_1^{2,j_2} & s_2^{2,j_2} \end{bmatrix}^T, \quad j_2 = m_2, m_2 + 1, \dots, m_2 + k_2.$$

Подстроки должны соответствовать следующим условиям

$$s_1^{1,m_1} = s_1^{2,m_2}, \quad (2.25)$$

$$T_{m_1}(m_1 + k_1) = 0, \quad T_{m_2}(m_2 + k_2) = 0. \quad (2.26)$$

После нахождения подстрок в строках (2.23) обмениваем их, оставляя неизменными первые элементы подстрок.

$$S_1 = \left(s^{1,1}, \dots, \underbrace{s^{1,m_1}, \dots, s^{1,m_1+k_1}}_{S_1(m_1)}, \dots, s^{1,L_1} \right), \quad (2.27)$$

$$S_2 = \left(s^{2,1}, \dots, \underbrace{s^{2,m_2}, \dots, s^{2,m_2+k_2}}_{S_2(m_2)}, \dots, s^{2,L_2} \right), \quad (2.28)$$

$$S_3 = (s^{1,1}, \dots, s^{1,m_1}, s^{2,m_2+1}, \dots, s^{2,m_2+k_2}, s^{1,m_1+k_1+1}, \dots, s^{1,L_1}), \quad (2.29)$$

$$S_4 = (s^{2,1}, \dots, s^{2,m_2}, s^{1,m_1+1}, \dots, s^{1,m_1+k_1}, s^{2,m_2+k_2+1}, \dots, s^{2,L_2}). \quad (2.30)$$

В результате получаем новые строки кодов

$$S_3 = (s^{3,1}, \dots, s^{3,L_3}), S_4 = (s^{4,1}, \dots, s^{4,L_4}), \quad (2.31)$$

где

$$L_3 = L_1 - k_1 + k_2, L_4 = L_2 - k_2 + k_1. \quad (2.32)$$

Для выполнения операции мутации находим любой код в строке $s^j = [s_1^j \ s_2^j]^T$ и заменяем случайно вторую компоненту кода

$$s_2^j = r \in \left\{ 1, \dots, \left| F_{s_1^j} \right| \right\}, \quad (2.33)$$

где $|F_\alpha|$ - мощность множества функций F_α с числом аргументов α .

Алгоритм для решения задачи синтеза системы управления методом генетического программирования включает следующие этапы:

0. Подготовительный этап. Создание базовых множеств переменных и параметров, а также множеств функций с одним двумя или тремя аргументами. В качестве переменных используются либо значения компонент вектора состояния объекта управления, либо значения отклонений координат объекта управления от терминальных условий. Вектор параметров ищется с помощью отдельного алгоритма в процессе решения задачи синтеза.
1. Генерация начального множества возможных решений из упорядоченных множеств кодов, состоящих из двух компонентных векторов. Для каждой компоненты вектора управления необходимо найти свое математическое выражение. Коды математических выражений должны иметь ограниченную длину и удовлетворять условиям правильного кодирования математического выражения (2.16), (2.17).
2. Генерация множеств кодов значений параметров, если для поиска значений параметров используется генетический алгоритм. Каждый параметр представляется в виде целочисленного двоичного кода Грея.

Подробное описание алгоритма поиска параметров с помощью генетического алгоритма будет приведено ниже в описании метода вариационного генетического программирования.

3. Вычисление значений функционалов для всех возможных решений из начального множества. Значения функционалов для каждого возможного решения сохраняются.
4. Вычисление рангов Парето для всех возможных решений и определения множества Парето для начального множества по нулевому значению ранга Парето.
5. Выполнение основного цикла, обеспечивающего эволюцию множества. В основном цикле выполняются для случайно отобранных двух кодов возможных решений генетические операции скрещивания и мутации. В результате выполнения этих операций получаются два новых решения.
6. Выполнение операций скрещивания для кодов значений параметров, соответствующих отобранным для скрещивания решениям, и получение двух новых значений векторов параметров для новых возможных решений.
7. Для каждого нового решения с новыми значениями параметров вычисляются значения функционалов и определяются ранги Парето.
8. Определение возможного решения с наибольшим рангом Парето. Если новое решение, полученное в результате выполнения операций скрещивания и мутации, имеет ранг Парето меньше наибольшего, то возможное решение с наибольшим рангом Парето заменяется новым возможным решением.

После выполнения всех циклов по эволюции множества возможных решений строится множество Парето по нулевому значению ранга Парето. Отбор возможного решения задачи синтеза управления осуществляет проектировщик из множества Парето по значениям функционалов.

2.3 Метод вариационного генетического программирования

Метод вариационного генетического программирования использует принцип малых вариаций базисного решения [11-14], который успешно применялся для решения различных задач синтеза управления в методе сетевого оператора [7,8,9].

Принцип малых вариаций базисного решения задает множество возможных решений в виде множества упорядоченных наборов вариаций одного возможного решения, называемого базисным.

$$G(S_0) = \{W_1, \dots, W_H\}, \quad (2.34)$$

где S_0 - код записи базисного решения,

$$S_0 = (s^{0,1}, \dots, s^{0,L_0}), \quad (2.35)$$

$s^{0,i} = [s_1^{0,i} \ s_2^{0,i}]^T$, $s_1^{0,i}$ - количество аргументов функции, $s_2^{0,i}$ - номер функции, $i = \overline{1, L_0}$, L_0 - длина кода записи базисного решения, $L_0 \leq L_+$, L_+ - заданная предельно возможная длина записи, W_i - упорядоченный набор вариаций,

$$W_i = (w^{i,1}, \dots, w^{i,l}), \quad i = \overline{1, H}, \quad (2.36)$$

$w^{i,j}$ - код описания малой вариации,

$$w^{i,j} = [w_1^{i,j} \ \dots \ w_v^{i,j}]^T, \quad i = \overline{1, H}, \quad j = \overline{1, l}, \quad (2.37)$$

l - длина набора вариаций, v - размерность вектора кода описания вариации.

Принцип малых вариаций базисного решения предполагает построение любого возможного решения в виде вариаций базисного решения.

$$G(S_0) = \{S_1, \dots, S_H\}, \quad (2.38)$$

где

$$S_i = W_i \circ S_0, \quad (2.39)$$

$$W_i \circ S_0 = w^{i,l} \circ \dots \circ w^{i,1} \circ S_0. \quad (2.40)$$

Длина l набора малых вариаций определяет максимальное количество вариаций между кодом возможного решения из множества (2.39) и кодом базисного решения, поэтому величина l играет роль расстояния между базисным

решением и элементом множества (2.39) возможных решений. В результате, если базисное решение подобрано разработчиком достаточно удачно, то величина l должна быть небольшой. При выполнении поиска оптимального решения и замене при поиске базисного решения на наилучшее возможное решение величину l целесообразно уменьшать по мере улучшения оценок наилучших возможных решений.

Малая вариация кода базисного решения должна обеспечивать малое изменение кода записи математического выражения, и при этом после каждой малой вариации код записи должен соответствовать правильному математическому выражению.

В методе вариационного генетического программирования используем пять малых вариаций записей кодов:

0 – изменение значения второй компоненты вектора кода элемента;

1 – удаление кода элемента с единичным значением первой компоненты;

2 – вставка в заданную позицию кода элемента с единичным значением первой компоненты;

3 – увеличение значения первой компоненты кода элемента на единицу и добавление после измененного элемента кода с нулевым значением первой компоненты;

4 – уменьшение значения первой компоненты кода элемента на единицу и удаление начинающегося со следующего элемента подвыражения.

Для представления малой вариации в вычислительных алгоритмах используем вектор из трех компонент

$$\mathbf{w} = [w_1 \ w_2 \ w_3]^T, \quad (2.41)$$

где w_1 - номер вариации, w_2 - номер варьируемого элемента или позиция для вставки нового элемента, w_3 - значение второго компонента.

Рассмотрим пример вариаций кодов математических выражений

Пусть заданы множества функций

$$F_0 = (x_1, x_2, q_1, q_2),$$

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

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

$$F_3 = (f_{3,1}(z_1, z_2, z_3), f_{3,2}(z_1, z_2, z_3)),$$

где

$$f_{3,1}(z_1, z_2, z_3) = \begin{cases} z_2, & \text{если } z_1 < 0 \\ z_3, & \text{иначе} \end{cases}, \quad f_{3,2}(z_1, z_2, z_3) = \begin{cases} z_1, & \text{если } z_3 > z_2 \\ z_2, & \text{иначе} \end{cases}.$$

Пусть задано математическое выражение базисного решения

$$y = \begin{cases} e^{-x_1} \sin(x_2), & \text{если } q_1 < q_2 \\ e^{-x_1} \cos(x_2), & \text{иначе} \end{cases} \quad (2.42)$$

или

$$y = e^{-x_1} f_{3,1}(q_1 - q_2, \sin(x_2), \cos(x_2)). \quad (2.43)$$

Запись кода базисного решения имеет вид

$$S_1 = \left(\begin{bmatrix} 2 \\ 2 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 3 \\ 1 \end{bmatrix}, \begin{bmatrix} 2 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 3 \end{bmatrix}, \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \begin{bmatrix} 0 \\ 4 \end{bmatrix}, \begin{bmatrix} 1 \\ 3 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 4 \end{bmatrix}, \begin{bmatrix} 0 \\ 2 \end{bmatrix} \right). \quad (2.44)$$

Пусть задан набор векторов вариаций из восьми элементов

$$W = (w^1, w^2, w^3, w^4, w^5, w^6, w^7, w^8).$$

Вектора вариаций в заданном наборе имеют следующие значения

$$W = \left(\begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 8 \\ 4 \end{bmatrix}, \begin{bmatrix} 2 \\ 8 \\ 3 \end{bmatrix}, \begin{bmatrix} 3 \\ 6 \\ 2 \end{bmatrix}, \begin{bmatrix} 4 \\ 5 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 5 \\ 2 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} \right).$$

Первый вектор вариаций $w^1 = [0 \ 1 \ 1]^T$ указывает, что необходимо сделать вариацию $w_1^1 = 0$, изменения значения второй компоненты элемента $w_2^1 = 1$ в записи на значение $w_3^1 = 1$. В результате получаем

$$S_2 = w^1 \circ S_1,$$

$$S_2 = \left(\begin{bmatrix} 2 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 3 \\ 1 \end{bmatrix}, \begin{bmatrix} 2 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 3 \end{bmatrix}, \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \begin{bmatrix} 0 \\ 4 \end{bmatrix}, \begin{bmatrix} 1 \\ 3 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 4 \end{bmatrix}, \begin{bmatrix} 0 \\ 2 \end{bmatrix} \right).$$

Новая запись соответствует математическому выражению

$$y_2 = e^{-x_1} + f_{3,1}(q_1 - q_2, \sin(x_1), \cos(x_2)).$$

Следующий вектор вариаций $\mathbf{w}^2 = [1 \ 8 \ 4]^T$ указывает, что выполняется вариация $w_1^2 = 1$, удаление элемента $w_2^2 = 8$ $\mathbf{s}^2 = [1 \ 2]^T$ из записи, если он имеет единичное значение первой компоненты $s_1^2 = 1$. Получаем

$$\mathbf{S}_3 = \mathbf{w}^2 \circ \mathbf{S}_2,$$

$$\mathbf{S}_3 = \left(\begin{array}{cccccccccccc} [2] & [1] & [1] & [0] & [3] & [2] & [0] & [0] & [1] & [0] & [1] & [0] \\ [1] & [1] & [2] & [1] & [1] & [1] & [3] & [4] & [3] & [1] & [4] & [2] \end{array} \right).$$

Полученная запись соответствует математическому выражению

$$y_3 = e^{-x_1} + f_{3,1}(q_1 + q_2, \sin(x_1), \cos(x_2)).$$

Следующий вектор вариации $\mathbf{w}^3 = [2 \ 8 \ 3]^T$ соответствует вставке в позицию $w_2^3 = 8$ кода элемента со значением первой компоненты 1 и второй компоненты $w_3^3 = 3$.

$$\mathbf{S}_4 = \mathbf{w}^3 \circ \mathbf{S}_3,$$

$$\mathbf{S}_4 = \left(\begin{array}{cccccccccccc} [2] & [1] & [1] & [0] & [3] & [2] & [0] & [1] & [0] & [1] & [0] & [1] & [0] \\ [1] & [1] & [2] & [1] & [1] & [1] & [3] & [3] & [4] & [3] & [1] & [4] & [2] \end{array} \right).$$

Запись соответствует следующему математическому выражению:

$$y_4 = e^{-x_1} + f_{3,1}(q_1 + \sin(q_2), \sin(x_1), \cos(x_2))$$

Следующий вектор вариации $\mathbf{w}^4 = [3 \ 6 \ 2]^T$ указывает на вариацию $w_1^4 = 3$, увеличения аргумента функции, код которой расположен в позиции $w_2^4 = 6$ и вставку после данного элемента нульместной функции под номером $w_3^4 = 2$.

После вариации получаем

$$\mathbf{S}_5 = \mathbf{w}^4 \circ \mathbf{S}_4,$$

$$\mathbf{S}_5 = \left(\begin{array}{cccccccccccc} [2] & [1] & [1] & [0] & [3] & [2] & [1] & [0] & [1] & [0] & [1] & [0] & [1] & [0] \\ [1] & [1] & [2] & [1] & [1] & [1] & [3] & [2] & [3] & [4] & [3] & [1] & [4] & [2] \end{array} \right).$$

Запись соответствует математическому выражению

$$y_5 = e^{-x_1} + f_{3,1}(\sin(x_2) + \sin(q_2), \sin(x_1), \cos(x_2)).$$

Следующий вектор вариаций $\mathbf{w}^5 = [4 \ 5 \ 1]^T$ указывает на вариацию $w_1^5 = 4$, состоящей в уменьшении на единицу аргумента функции, соответствующей элементу $w_2^5 = 5$. Для корректного выполнения вариации необходимо найти подстроку, правильно описывающую математическое выражение, начиная с элемента 6, и удалить данную подстроку. После выполнения вариации получаем

$$\tilde{\mathbf{S}}_6 = \mathbf{w}^5 \circ \mathbf{S}_5,$$

$$\tilde{\mathbf{S}}_6 = \left(\begin{array}{cccccccccccccccc} [2] & [1] & [1] & [0] & [2] & [2] & [1] & [0] & [1] & [0] & [1] & [0] & [1] & [0] \\ [1] & [1] & [2] & [1] & [1] & [1] & [3] & [2] & [3] & [4] & [3] & [1] & [4] & [2] \end{array} \right).$$

Код $\tilde{\mathbf{S}}_6$ не соответствует правильной записи математического выражения.

Находим подстроку, начиная с элемента $\tilde{\mathbf{s}}^6$ по формуле (2.20), пока не получим нулевое значение индекса

$$T_6(6) = \tilde{\mathbf{s}}_1^6 = 2,$$

$$T_6(7) = 2 - 1 + 1 = 2,$$

$$T_6(8) = 2 - 2 + 1 + 0 = 1,$$

$$T_6(9) = 2 - 3 + 1 + 0 + 1 = 1,$$

$$T_6(10) = 2 - 4 + 1 + 0 + 1 + 0 = 0.$$

В результате получаем, что подстрока содержит элементы $\mathbf{S}_6(6) = \tilde{\mathbf{s}}^6, \dots, \tilde{\mathbf{s}}^{10}$

$$\tilde{\mathbf{S}}_6 = \left(\begin{array}{cccccccccccccccc} [2] & [1] & [1] & [0] & [2] & [2] & [1] & [0] & [1] & [0] & [1] & [0] & [1] & [0] \\ [1] & [1] & [2] & [1] & [1] & [1] & [3] & [2] & [3] & [4] & [3] & [1] & [4] & [2] \end{array} \right).$$

$\underbrace{\hspace{10em}}_{\mathbf{S}_6(6)}$

Удаляем найденную подстроку, получаем правильный код записи математического выражения

$$\mathbf{S}_6 = \left(\begin{array}{cccccccc} [2] & [1] & [1] & [0] & [2] & [1] & [0] & [1] & [0] \\ [1] & [1] & [2] & [1] & [1] & [3] & [1] & [4] & [2] \end{array} \right).$$

Код соответствует следующему математическому выражению:

$$y_6 = e^{-x_1} + \sin(x_1) + \cos(x_2).$$

Следующий вектор вариации $\mathbf{w}^6 = [1 \ 5 \ 2]^T$ описывает вариацию удаления элемента, соответствующего функции с одним аргументом. В записи кода данный элемент $\mathbf{s}^5 = [2 \ 1]^T$ указывает на код функции с количеством аргументов $s_1^5 = 2$, поэтому данная вариация не выполняется

$$S_7 = S_6,$$

$$y_7 = y_6 = e^{-x_1} + \sin(x_1) + \cos(x_2).$$

Следующий вектор вариации $\mathbf{w}^7 = [2 \ 7 \ 3]^T$ описывает вариацию вставки элемента, соответствующего функции с одним аргументом под номером $w_3^7 = 3$.

В результате вариации получаем

$$S_8 = \mathbf{w}^7 \circ S_7,$$

$$S_8 = \left(\left[\begin{array}{c} 2 \\ 1 \end{array} \right], \left[\begin{array}{c} 1 \\ 1 \end{array} \right], \left[\begin{array}{c} 1 \\ 2 \end{array} \right], \left[\begin{array}{c} 0 \\ 1 \end{array} \right], \left[\begin{array}{c} 2 \\ 1 \end{array} \right], \left[\begin{array}{c} 1 \\ 3 \end{array} \right], \left[\begin{array}{c} 1 \\ 3 \end{array} \right], \left[\begin{array}{c} 0 \\ 1 \end{array} \right], \left[\begin{array}{c} 1 \\ 4 \end{array} \right], \left[\begin{array}{c} 0 \\ 2 \end{array} \right] \right).$$

Полученный код записи соответствует математическому выражению

$$y_8 = e^{-x_1} + \sin(\sin(x_1)) + \cos(x_2).$$

Последний в наборе вектор вариации $\mathbf{w}^8 = [0 \ 5 \ 2]^T$ указывает на замену второй компоненты элемента $w_2^8 = 5$ на значение $w_3^8 = 2$. В результате получаем

$$S_9 = \mathbf{w}^8 \circ S_8,$$

$$S_9 = \left(\left[\begin{array}{c} 2 \\ 1 \end{array} \right], \left[\begin{array}{c} 1 \\ 1 \end{array} \right], \left[\begin{array}{c} 1 \\ 2 \end{array} \right], \left[\begin{array}{c} 0 \\ 1 \end{array} \right], \left[\begin{array}{c} 2 \\ 2 \end{array} \right], \left[\begin{array}{c} 1 \\ 3 \end{array} \right], \left[\begin{array}{c} 1 \\ 3 \end{array} \right], \left[\begin{array}{c} 0 \\ 1 \end{array} \right], \left[\begin{array}{c} 1 \\ 4 \end{array} \right], \left[\begin{array}{c} 0 \\ 2 \end{array} \right] \right).$$

Итоговое математическое выражение имеет следующий вид:

$$y_9 = e^{-x_1} + \sin(\sin(x_1))\cos(x_2).$$

Основным преимуществом метода малых вариаций базисного решения является возможность при поиске использовать аналитические методы и интуитивные соображения при построении базисного решения. Другим важным преимуществом метода поиска, основанного на использовании принципа малых вариаций, является применение генетических операций не к коду записи

математического выражения, а к наборам векторов вариаций, при этом не требуется выполнения проверки кода записи математического выражения.

Основным преимуществом метода вариационного генетического программирования перед методом сетевого оператора является возможность использования при синтезе произвольный набор функций с любым количеством аргументов. Метод сетевого оператора использует только функции с одним или двумя аргументами.

При синтезе системы управления используем множества параметров и переменных:

$$F_0 = (x_1, \dots, x_n, q_1, \dots, q_p), \quad (2.45)$$

и следующие множества из функций с одним, двумя и тремя аргументами

$$\begin{aligned} F_1 = & (f_{1,0}(z) = z, f_{1,1}(z) = \operatorname{sgn}(z)z^2, f_{1,2}(z) = z^2, f_{1,3}(z) = -z, f_{1,4}(z) = \operatorname{sgn}(z)\sqrt{|z|}, \\ & f_{1,5}(z) = z^{-1}, f_{1,6}(z) = e^z, f_{1,7}(z) = \ln|z|, f_{1,8}(z) = \frac{1 - e^{-z}}{1 + e^{-z}}, f_{1,9}(z) = \begin{cases} 1, & \text{если } z \geq 0 \\ 0, & \text{иначе} \end{cases}, \\ & f_{1,10}(z) = \operatorname{sgn}(z), f_{1,11}(z) = \cos(z), f_{1,12}(z) = \sin(z), f_{1,13}(z) = \arctan(z), f_{1,14}(z) = z^3, \\ & f_{1,15}(z) = \sqrt[3]{z}, f_{1,16}(z) = \begin{cases} z, & \text{если } |z| \leq 1 \\ \operatorname{sgn}(z), & \text{иначе} \end{cases}, f_{1,17}(z) = \operatorname{sgn}(z)\ln(|z| + 1), \\ & f_{1,18}(z) = \operatorname{sgn}(z)(e^{-|z|} - 1), f_{1,19}(z) = \operatorname{sgn}(z)(1 - e^{-|z|}), f_{1,20}(z) = \begin{cases} 1, & \text{если } z > a \\ -1, & \text{если } z < 0 \\ 3za^{-2} - 2z^3a^{-3}, & \text{иначе} \end{cases} \\ & , f_{1,21}(z) = \begin{cases} 1, & \text{если } z > a/2 \\ -1, & \text{если } z < -a/2 \\ 3za^{-2} - 4z^3a^{-3}, & \text{иначе} \end{cases}, \\ & f_{1,22}(z) = (z - z^3), \end{aligned} \quad (2.46)$$

$$F_2 = (f_{2,0}(z_1, z_2) = z_1 + z_2, f_{2,1}(z_1, z_2) = z_1 z_2, f_{2,2}(z_1, z_2) = \max\{z_1, z_2\},$$

$$f_{2,3}(z_1, z_2) = \min\{z_1, z_2\}, f_{2,4}(z_1, z_2) = z_1 + z_2 - z_1 z_2,$$

$$f_{2,5}(z_1, z_2) = \operatorname{sgn}(z_1 + z_2)\sqrt{z_1^2 + z_2^2}, f_{2,6}(z_1, z_2) = \operatorname{sgn}(z_1 + z_2)(|z_1| + |z_2|),$$

$$f_{2,7}(z_1, z_2) = \text{sgn}(z_1 + z_2) \max\{|z_1|, |z_2|\}, \quad (2.47)$$

$$F_3 = \left(f_{3,0}(z_1, z_2, z_3) = \begin{cases} z_2, & \text{если } z_1 \leq 0 \\ z_3, & \text{иначе} \end{cases}, f_{3,1}(z_1, z_2, z_3) = \begin{cases} z_3, & \text{если } z_1 \leq z_2 \\ -z_3, & \text{иначе} \end{cases}, \right.$$

$$f_{3,2}(z_1, z_2, z_3) = \begin{cases} z_2, & \text{если } z_1 < z_2 \\ z_3, & \text{если } (z_1 \geq z_2) \wedge (z_1 > z_3), \\ z_1, & \text{если } (z_1 \geq z_2) \wedge (z_1 \leq z_3) \end{cases},$$

$$f_{3,3}(z_1, z_2, z_3) = \begin{cases} z_2 - z_1, & \text{если } |z_1 - z_2| < |z_1 - z_3| \\ z_3 - z_1, & \text{иначе} \end{cases}, \quad (2.48)$$

При решении задачи синтеза системы управления в алгоритме метода вариационного генетического программирования предусмотрен одновременный поиск структуры в виде кода записи (2.13) и вектора параметров $\mathbf{q} = [q_1 \dots q_p]^T$.

Рассмотрим основные этапы алгоритма. Для формирования начального множества возможных решений задаем код базисного решения

$$\mathbf{S}^0 = (\mathbf{s}^{0,1}, \dots, \mathbf{s}^{0,L_0}) \quad (2.49)$$

и множество наборов векторов вариаций

$$\mathbf{G} = \{\mathbf{W}_0, \mathbf{W}_1, \dots, \mathbf{W}_H\}, \quad (2.50)$$

где

$$\mathbf{W}_i = (\mathbf{w}^{i,1}, \dots, \mathbf{w}^{i,l}), \quad i = \overline{0, H}. \quad (2.51)$$

Чтобы базисное решение участвовало в поиске оптимального решения во множество возможных решений (2.49) включен набор

$$\mathbf{W}_0 = \left(\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \dots, \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \right). \quad (2.52)$$

Считаем, что вектор вариаций из нулевых компонент не выполняет никакую вариацию

$$\mathbf{S}^i = [0 \ 0 \ 0]^T \circ \mathbf{S}^i. \quad (2.53)$$

Одновременно генерируем множества кодов Грея для поиска значений векторов параметров

$$\mathbf{Y} = \{\mathbf{y}^0, \mathbf{y}^1, \dots, \mathbf{y}^H\}, \quad (2.54)$$

где

$$\mathbf{y}^i = (y_1^i, \dots, y_{c+d}^i, y_{c+d+1}^i, \dots, y_{(c+d)p}^i), \quad y_j^i \in \{0,1\}, \quad j = \overline{1, p(c+d)}, \quad i = \overline{0, H}, \quad (2.55)$$

c - число бит под целую часть значения параметра, d - число бит под дробную часть значения параметра.

Для получения значения вектора параметра q_k^i по его коду Грея $(y_{(k-1)(c+d)+1}^i, \dots, y_{k(c+d)}^i)$, $1 \leq k \leq p$, $0 \leq i \leq H$, используем следующие соотношения:

$$q_k^i = \sum_{j=1}^{c+d} 2^{c-1-j} b_j, \quad (2.56)$$

где

$$b_1 = y_{(k-1)(c+d)+1}^i, \quad b_j = b_{j-1} \oplus y_{(k-1)(c+d)+j}^i, \quad j = \overline{2, c}, \quad (2.57)$$

$$b_{c+1} = y_{(k-1)(c+d)+c+1}^i, \quad b_j = b_{j-1} \oplus y_{(k-1)(c+d)+c+j}^i, \quad j = \overline{c+2, c+d}, \quad (2.58)$$

\oplus - операция суммирования по модулю 2, $\alpha \oplus \beta = (\alpha + \beta) \bmod 2$.

Любое возможное решение определяется парой элементов (W_i, \mathbf{y}^i) из множеств G и Y . Элемент W_i совместно с базисным решением (2.49) S^0 определяет структуру математического выражения, а элемент – вектор параметров \mathbf{q}^i , входящих в это математическое выражение.

Для каждого возможного решения (W_i, \mathbf{y}^i) , $i \in \{0, \dots, H\}$, подсчитываем значения функционалов (1.19), (1.20). Получаем множество значений векторов функционалов

$$\Xi = \{ \mathbf{J}^0, \mathbf{J}^1, \dots, \mathbf{J}^H \}, \quad (2.59)$$

где $\mathbf{J}^i = [J_1^i \ J_2^i]^T$, $i = \overline{0, H}$.

Далее подсчитываем для каждого возможного решения (W_i, \mathbf{y}^i) , $i \in \{0, \dots, H\}$ ранги Парето $\Lambda(i)$ по формуле

$$\Lambda(i) = \sum_{k=0}^H \lambda_i(k), \quad (2.60)$$

где

$$\lambda_i(k) = \begin{cases} 1, & \text{если } \mathbf{J}^k \leq \mathbf{J}^i \\ 0, & \text{иначе} \end{cases}. \quad (2.61)$$

В процессе эволюции множества возможных решений выполняем генетические операции скрещивания и мутации. Отбираем для скрещивания случайно два возможных решения $(\mathbf{W}_{i_1}, \mathbf{y}^{i_1})$, $(\mathbf{W}_{i_2}, \mathbf{y}^{i_2})$, $i_1, i_2 \in \{0, \dots, H\}$.

Определяем для них вероятность скрещивания

$$p_c = \max \left\{ \frac{1 + \gamma \Lambda(i_1)}{1 + \Lambda(i_1)}, \frac{1 + \gamma \Lambda(i_2)}{1 + \Lambda(i_2)} \right\}, \quad (2.62)$$

где $\gamma < 1$, параметр, определяющий наименьшую вероятность скрещивания.

Из формулы (2.62) следует, что наибольшая вероятность скрещивания достигается для возможного решения $(\mathbf{W}_i, \mathbf{y}^i)$ с рангом Парето $\Lambda(i) = 1$. Так как ранг Парето не превосходит количества элементов во множестве возможных решений, то наименьшую вероятность скрещивания определяем неравенством

$$p_c > \frac{1 + \gamma H}{1 + H} \approx \gamma. \quad (2.63)$$

Для операции скрещивания находим точки скрещивания. Скрещивание выполняем отдельно для структурной и параметрической частей возможных решений

После выполнения скрещивания получаем четыре новых решения, потомка

$$\begin{aligned} & (\mathbf{W}_{H+1}, \mathbf{y}^{H+1}), (\mathbf{W}_{H+2}, \mathbf{y}^{H+2}), (\mathbf{W}_{H+3}, \mathbf{y}^{H+3}), (\mathbf{W}_{H+4}, \mathbf{y}^{H+4}) \\ \mathbf{W}_{H+1} &= (\mathbf{w}^{i_1, 1}, \dots, \mathbf{w}^{i_1, k_s}, \mathbf{w}^{i_2, k_s+1}, \dots, \mathbf{w}^{i_2, l}), \end{aligned} \quad (2.64)$$

$$\mathbf{y}^{H+1} = \left(y_1^{i_1}, \dots, y_{k_p}^{i_1}, y_{k_p+1}^{i_2}, \dots, y_{(c+d)p}^{i_2} \right), \quad (2.65)$$

$$\mathbf{W}_{H+2} = (\mathbf{w}^{i_2, 1}, \dots, \mathbf{w}^{i_2, k_s}, \mathbf{w}^{i_1, k_s+1}, \dots, \mathbf{w}^{i_1, l}), \quad (2.66)$$

$$\mathbf{y}^{H+2} = \left(y_1^{i_2}, \dots, y_{k_p}^{i_2}, y_{k_p+1}^{i_1}, \dots, y_{(c+d)p}^{i_1} \right), \quad (2.67)$$

$$\mathbf{W}_{H+3} = \mathbf{W}_{i_1}, \quad (2.68)$$

$$\mathbf{y}^{H+3} = \mathbf{y}^{H+1}, \quad (2.69)$$

$$\mathbf{W}_{H+4} = \mathbf{W}_{i_2}, \quad (2.70)$$

$$\mathbf{y}^{H+4} = \mathbf{y}^{H+2}, \quad (2.71)$$

где k_s, k_p - точки скрещивания, $k_s \in \{1, \dots, l\}$, $k_p \in \{1, \dots, p(c+d)\}$.

Далее в полученных решениях $(\mathbf{W}_{H+1}, \mathbf{y}^{H+1})$, $(\mathbf{W}_{H+2}, \mathbf{y}^{H+2})$, $(\mathbf{W}_{H+3}, \mathbf{y}^{H+3})$, $(\mathbf{W}_{H+4}, \mathbf{y}^{H+4})$ выполняем с заданной вероятностью p_μ операцию мутации.

Находим точки мутации $k_\mu, k_\eta, k_\mu \in \{1, \dots, l\}$, $k_\eta \in \{1, \dots, p(c+d)\}$. В каждом из решений случайно заменяем вектор вариации \mathbf{w}^{H+i, k_μ} и компоненту кода для вектора параметров $y_{k_\eta}^{H+i} \in \{0, 1\}$, $i = \overline{1, 4}$.

Далее анализируем полученные решения на предмет их включения во множество возможных решений.

Анализ производим последовательно для каждого нового решения.

Вычисляем для первого нового решения $(\mathbf{W}_{H+1}, \mathbf{y}^{H+1})$ значения функционалов (1.19), (1.20), получаем $\mathbf{J}^{H+1} = [J_1^{H+1} \ J_2^{H+1}]^T$. Вычисляем для этого решения ранг Парето $\Lambda(H+1)$.

Определяем во множестве возможных решений решение $(\mathbf{W}_{i+}, \mathbf{y}^{i+})$ с наибольшим рангом Парето

$$\Lambda(i+) = \max\{\Lambda(0), \dots, \Lambda(H)\}. \quad (2.72)$$

Сравниваем ранг Парето $\Lambda(H+1)$ нового возможного решения с наибольшим рангом Парето $\Lambda(i+)$.

Если выполняются условия

$$\Lambda(H+1) < \Lambda(i+), \quad (2.73)$$

то заменяем возможное решение с наибольшим рангом Парето на новое полученное решение

$$\left(\mathbf{W}_{i^+}, \mathbf{y}^{i^+} \right) \leftarrow \left(\mathbf{W}_{H+1}, \mathbf{y}^{H+1} \right), \quad (2.74)$$

$$\mathbf{J}^{i^+} \leftarrow \mathbf{J}^{H+1}. \quad (2.75)$$

Если замена (2.74), (2.75) выполнялась, то пересчитываем ранги Парето для всех возможных решений из множества по формулам (2.60), (2.61).

Повторяем процесс анализа включения для остальных новых полученных решений $\left(\mathbf{W}_{H+2}, \mathbf{y}^{H+2} \right), \left(\mathbf{W}_{H+3}, \mathbf{y}^{H+3} \right), \left(\mathbf{W}_{H+4}, \mathbf{y}^{H+4} \right)$.

После выполнения заданного количества циклов скрещивания для полученного на данный момент множества строим множество Парето по нулевым значениям рангов Парето. Из множества Парето отбираем по установленному для конкретной задачи критерию наилучшее решение и меняем им базисное решение.

2.4 Пример синтеза управления методами генетического программирования и вариационного генетического программирования

Проведем численный эксперимент для сравнения методов генетического программирования и вариационного генетического программирования в задачи синтеза системы управления.

Рассмотрим следующую задачу синтеза управления [12].

Задана модель объекта управления

$$\dot{x}_1 = x_1 - x_1 x_2 + a x_1 u, \quad (2.76)$$

$$\dot{x}_2 = x_2 + x_1 x_2 - b x_2 u, \quad (2.77)$$

$$\dot{x}_3 = (x_1 - 1)^2 + (x_2 - 1)^2. \quad (2.78)$$

Задано множество начальных значений

$$\mathbf{X}_0 = \left\{ x_1(0) \in [x_1^-, x_1^+], x_2(0) \in [x_2^-, x_2^+], x_3(0) = 0 \right\}. \quad (2.79)$$

Заданы ограничения на управление

$$u^- \leq u \leq u^+. \quad (2.80)$$

Задан критерий качества

$$J = x_3(t_f) \rightarrow \min, \quad (2.81)$$

где t_f - заданная положительная величина.

Необходимо найти управление в виде функции

$$u = h(x_1, x_2, x_3). \quad (2.82)$$

Заменяем множество начальных значений множеством точек начальных значений

$$X_0 = \{x_i(0) \in (x_i^- + j\Delta x_i : j = \overline{0, k_i}), i = 1, 2, x_3(0) = 0\}, \quad (2.83)$$

где $k_i + 1$ - заданное число точек начальных значений,

$$\Delta x_i = \frac{x_i^+ - x_i^-}{k_i}, \quad i = 1, 2. \quad (2.84)$$

Заменяем критерий качества суммой критериев, вычисленных для каждого начального значения

$$\bar{J} = \sum_{i=0}^{k_1} \sum_{j=0}^{k_2} (x_3(t_f))_{x_{1,i}, x_{2,j}} \rightarrow \min, \quad (2.85)$$

где $x_{1,i} = x_1^- + i\Delta x_1$, $i = \overline{0, k_1}$, $x_{2,j} = x_2^- + j\Delta x_2$, $j = \overline{0, k_2}$.

Задачу решаем методами генетического программирования и вариационного генетического программирования.

Для метода вариационного генетического программирования задаем базисное решение в форме тривиального соотношения

$$u = q_1 x_1 + q_2 x_2 + q_3 x_3, \quad (2.86)$$

где $q_1 = 1$, $q_2 = 1$, $q_3 = 1$.

Во время вычислительного эксперимента параметры модели имели следующие значения: $a = 0,4$, $b = 0,2$, $t_f = 12$ с, $x_1^- = 0,5$, $x_1^+ = 0,6$, $x_2^- = 0,7$, $x_2^+ = 0,8$, $k_1 = 2$, $k_2 = 2$. Шаг интегрирования составлял 0,005 с.

Для обоих методов были заданы одинаковые параметры генетического алгоритма: число возможных решений в начальном множестве 256, число

поколений 128, число возможных скрещиваний в одном поколении 128, вероятность мутации 0,7, длина записи составляла 32 двухкомпонентных вектора.

Для метода вариационного генетического программирования при получении каждого решения использовалось 8 вариаций базисного решения.

При синтезе полученные управления ограничивались согласно соотношению

$$u = \begin{cases} 0, & \text{если } \tilde{u} < 0 \\ 1, & \text{если } \tilde{u} > 1. \\ \tilde{u} & - \text{иначе} \end{cases}$$

Метод генетического программирования получил следующее решение

$$\tilde{u} = \begin{cases} Aq_2x_2 - \cos(x_1), & \text{если } |\cos(x_1) - Aq_2x_2| < |\cos(x_1) - x_3|, \\ x_3 - \cos(x_1), & \text{иначе} \end{cases},$$

где

$$A = \begin{cases} q_3, & \text{если } \sqrt[3]{B} < q_2 + q_3 - q_2q_3, \\ -q_3 & - \text{иначе} \end{cases},$$

$$B = \begin{cases} x_2 - \min\{q_3, x_2\}, & \text{если } |\min\{q_3, x_2\} - x_2| < |\min\{q_3, x_2\} - x_1|, \\ x_1 - \min\{q_3, x_2\}, & \text{иначе} \end{cases},$$

$$q_1 = 7,11670, \quad q_2 = 10,39893, \quad q_3 = 1,75122.$$

Метод вариационного генетического программирования получил следующее решение:

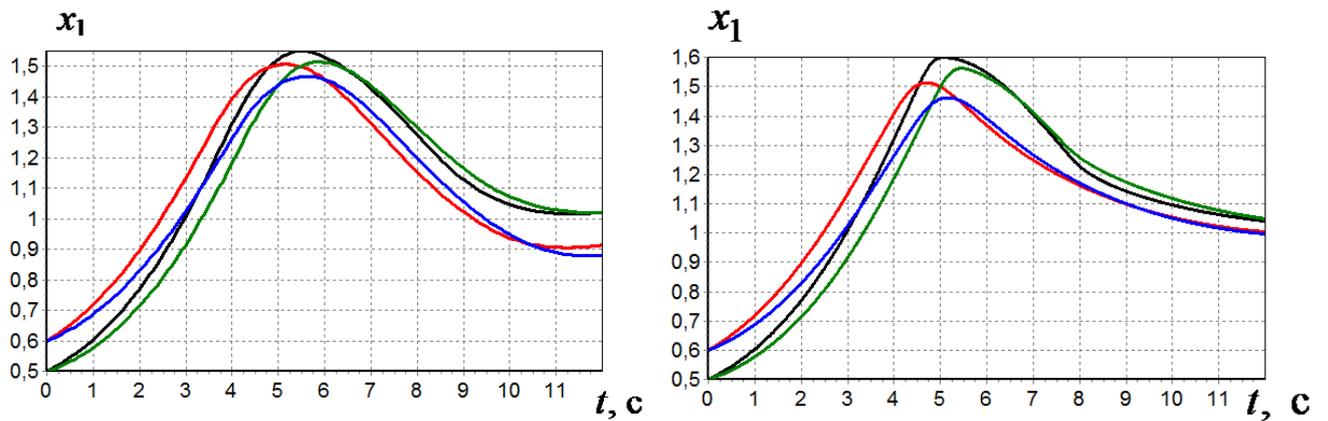
$$\tilde{u} = \operatorname{sgn} \left(\arctan(x_1^2 + \sin(\ln|\arctan(x_2)|)q) \right)^2 \times \\ \arctan(x_1^2 + \sin(\ln|\arctan(x_2)|)q)^2 (x_3 + x_2^2)x_1,$$

где $q = 3,23022$.

Вычисления выполнялись на компьютере с процессором Intel(R) Core(TM) i7-2640M CPU @ 2.80GHz 2.80 GHz. Процесс поиска для методов вариационного генетического программирования составил 5 мин. Поиск решения методом генетического программирования составил более 35 мин. На рисунок 2.7 - 2.10 приведены графики результатов моделирования полученных систем управления

при различных начальных условиях: $\mathbf{x}(0) = [0,5 \ 0,7 \ 0]^T$, $\mathbf{x}(0) = [0,6 \ 0,7 \ 0]^T$,
 $\mathbf{x}(0) = [0,5 \ 0,8 \ 0]^T$, $\mathbf{x}(0) = [0,6 \ 0,8 \ 0]^T$.

Как видно из результатов моделирования формы изменения управлений, полученных методами генетического и вариационного генетического программирования, существенно не отличаются. Критерий качества $x_3(t_f)$ при всех начальных значениях в системе управления, полученной методом вариационного генетического программирования, имеет меньшее значение, чем критерий для системы управления, полученной методом генетического программирования.



а — метод генетического программирования,
 б — метод вариационного генетического программирования.

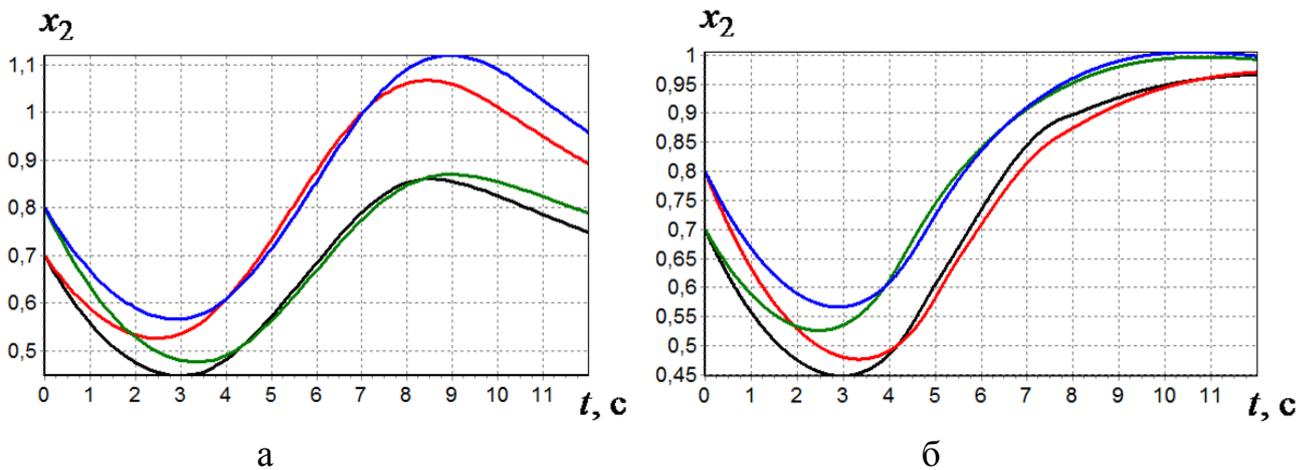


Рисунок 2.8. Графики изменения $x_2(t)$, а – метод генетического программирования,
б – метод вариационного генетического программирования.

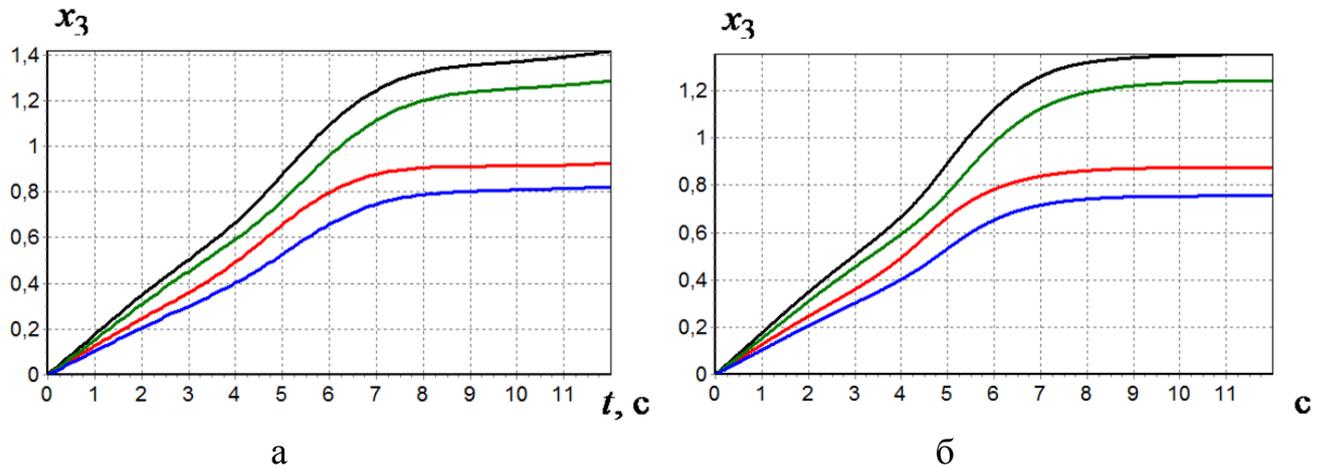


Рисунок 2.9. Графики изменения $x_3(t)$, а – метод генетического программирования,
б – метод вариационного генетического программирования.

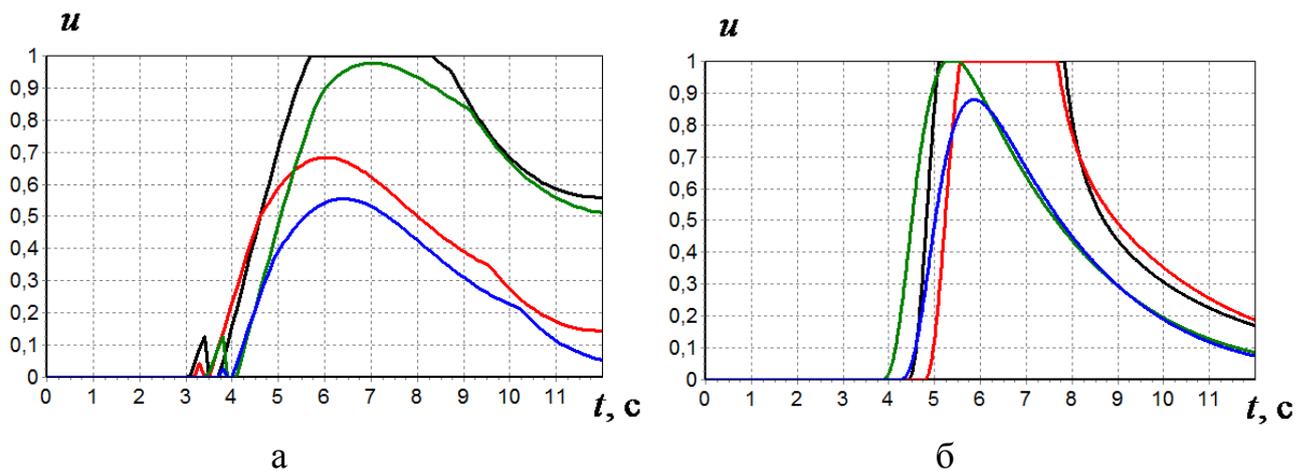


Рисунок 2.10. Графики изменения $u(t)$, а – метод генетического программирования,
б – метод вариационного генетического программирования.

2.5 Выводы к главе 2

1. Разработан метод вариационного генетического программирования для решения задачи синтеза системы управления. Метод включает основные преимущества известных методов. Метод вариационного генетического программирования использует принцип малых вариаций базисного решения, заимствованный из метода сетевого оператора и представление математического выражение в форме строки кодов элементов математического выражения.
2. Разработан метод кодирования математических выражений в форме числовых двухкомпонентных векторов. Первый компонент вектора указывает на число аргументов кодируемой функции. Второй компонент вектора указывает на номер функции из заданного множества функций.
3. Разработан метод поиска оптимального решения для задачи синтеза системы на основе принципа малых вариаций базисного решения. Определены малые вариации кода записи математического выражения. Метод вариационного генетического программирования использует четыре малых вариации кода.
4. Проведен вычислительный эксперимент по сравнению методов генетического программирования и вариационного программирования на примере решения задачи синтеза управления для тестовой системы «хищник-жертва». Вычисления показали, что даже для типового базисного, не учитывающего особенности задачи, метод вариационного генетического программирования при одинаковых параметрах генетического алгоритма работает в семь раз быстрее и находит решения, которое дают лучшие значения критерия качества, чем метод генетического программирования.

3. СИНТЕЗ СИСТЕМЫ УПРАВЛЕНИЯ МОБИЛЬНЫМ РОБОТОМ МЕТОДОМ ВАРИАЦИОННОГО ГЕНЕТИЧЕСКОГО ПРОГРАММИРОВАНИЯ

Рассмотрим решение задачи синтеза системы управления мобильным роботом методов вариационного генетического программирования [13]. В задаче необходимо найти синтезирующую функцию, описывающую зависимость значений вектора управления мобильным роботом от значений координат его состояния, с целью обеспечения перемещения робота в заданное терминальное состояние. Согласно задаче управление должно обеспечить перемещение мобильного робота заданное терминальное состояние из различных начальных условий, в процессе движения мобильные не должен заехать в ограничивающие области. Для робота известны его габаритные размеры и координаты углов, определяющих ограничивающие области.

На рисунок 3.1 приведена схема управления и геометрические параметры мобильного робота [13,14].

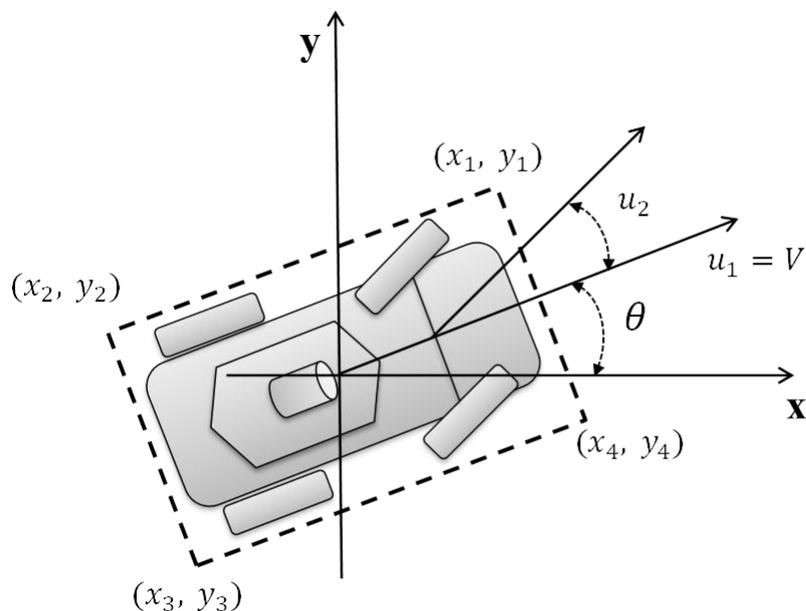


Рисунок 3.1 Схема управления мобильным роботом

Мобильный робот имеет две компоненты вектора управления, первая компонента определяет скорость движения мобильного робота, вторая компонента управления определяет угол направления движения робота.

Критерием качества управления является время перемещения робота в терминальное состояние.

Рассмотрим математическую формулировку рассматриваемой задачи синтеза системы управления.

3.1 Математическая модель объекта управления

Математическая модель объекта управления имеет следующий вид [13]:

$$\dot{x} = u_1 \cos(\theta), \quad (3.1)$$

$$\dot{y} = u_1 \sin(\theta), \quad (3.2)$$

$$\dot{\theta} = \frac{u_1}{L_b} \tan(u_2), \quad (3.3)$$

где x, y - координаты центра масс мобильного робота, θ - угол направления вектора скорости, L_b - обобщенный габаритный параметр мобильного робота.

На рисунок 3.1 $x_i, y_i, i = \overline{1,4}$, обозначают координаты углов области, определяющей габариты мобильного робота, x_1, y_1 - координаты переднего левого угла, x_2, y_2 - координаты заднего левого угла, x_3, y_3 - координаты заднего правого угла, x_4, y_4 - координаты переднего правого угла.

Положение углов робота определяются с помощью соотношений

$$x_i = x'_i \cos(\theta) - y'_i \sin(\theta), i = \overline{1,4}, \quad (3.4)$$

$$y_i = x'_i \sin(\theta) + y'_i \cos(\theta), i = \overline{1,4}, \quad (3.6)$$

$$x'_1 = x \cos(\theta) + y \sin(\theta) + L_b, \quad (3.7)$$

$$y'_1 = -x \sin(\theta) + y \cos(\theta) + \frac{L_b}{2}, \quad (3.8)$$

$$x'_2 = x \cos(\theta) + y \sin(\theta) - L_b, \quad (3.9)$$

$$y'_2 = -x \sin(\theta) + y \cos(\theta) + \frac{L_b}{2}, \quad (3.10)$$

$$x'_3 = x \cos(\theta) + y \sin(\theta) - L_b, \quad (3.11)$$

$$y'_3 = -x \sin(\theta) + y \cos(\theta) - \frac{L_b}{2}, \quad (3.12)$$

$$x'_4 = x \cos(\theta) + y \sin(\theta) + L_b, \quad (3.13)$$

$$y'_4 = -x \sin(\theta) + y \cos(\theta) - \frac{L_b}{2}. \quad (3.14)$$

Заданы ограничения на значения управлений

$$V^- \leq u_1 \leq V^+, \quad (3.15)$$

$$u_2^- \leq u_2 \leq u_2^+, \quad (3.16)$$

Задано множество начальных условий

$$X_0 = \{(x_{0,1}, y_{0,1}, \theta_{0,1}), \dots, (x_{0,M}, y_{0,M}, \theta_{0,M})\}. \quad (3.17)$$

Задано терминальное многообразие

$$x - x^f = 0, \quad (3.18)$$

$$y - y^f = 0, \quad (3.19)$$

$$\theta - \theta^f = 0. \quad (3.20)$$

где x^f , y^f , θ^f - координаты терминального положения центра масс робота.

Заданы пространственные ограничения в виде логических функций

$$\left((x_j - x^{\alpha_i}) < 0 \right) \vee \left((x_j - x^{\beta_i}) > 0 \right) \wedge \left((y_j - y^{\alpha_i}) < 0 \right) \vee \left((y_j - y^{\beta_i}) > 0 \right), \quad j = \overline{1,4} \quad (3.21)$$

где

$$x^{\alpha_i} < x^{\beta_i}, \quad j = \overline{1,4} \quad (3.22)$$

$$y^{\alpha_i} < y^{\beta_i}, \quad j = \overline{1,4}, \quad i = \overline{1, k_p}, \quad (3.23)$$

k_p - число ограничивающих областей или число препятствий.

Задача заключается в том, чтобы управление в форме функций от координат пространства состояний

$$u_1 = h_1(x, y, \theta), \quad (3.24)$$

$$u_2 = h_2(x, y, \theta). \quad (3.25)$$

Управление (3.24), (3.25) должно обеспечивать перемещение робота из любого заданного в (3.17) начального состояния в терминальное положение (3.18) – (3.20), удовлетворяя при этом ограничениям (3.21) для всех углов робота за минимальное время

$$J_1 = t_f \rightarrow \min, \quad (3.26)$$

где t_f - время попадания центром масс робота в терминальное состояние (3.18) – (3.20).

Для решения задачи используем вариационный генетический алгоритм. Рассмотрим решение задачи при следующих значениях параметров: $L_b = 4$, $V^- = -5$, $V^+ = 5$, $u^- = -1$, $u^+ = 1$, $k_p = 2$, $x^{\alpha_1} = -20$, $y^{\alpha_1} = -2$, $x^{\beta_1} = -6$, $y^{\beta_1} = 2$, $x^{\alpha_2} = 6$, $y^{\alpha_2} = -2$, $x^{\beta_2} = 20$, $y^{\beta_2} = 2$.

Множество начальных значений имело следующий вид:

$$X_0 = \{(-8, -4, 0), (8, -4, 0), (-8, 4, 0), (8, 4, 0)\}. \quad (3.27)$$

Терминальные условия имели значения: $x^f = 0$, $y^f = 0$, $\theta^f = 0$.

Множество аргументов включало следующие элементы

$$F_0 = (x^f - x, y^f - y, \theta^f - \theta, q_1, q_2, q_3), \quad (3.28)$$

где q_1, q_2, q_3 - числовые параметры, которые также ищутся вместе со структурами функций (3.24), (3.25).

При решении задачи синтеза методом вариационного генетического программирования используем наборы функций [11,13,14] с одним, двумя и тремя аргументами, F_1, F_2, F_3 (2.46) – (2.48), приведенные в разделе 2.

Задаем базисное решение в виде

$$u_i = \begin{cases} u_i^-, & \text{если } \tilde{u}_i \leq u_i^- \\ u_i^+, & \text{если } \tilde{u}_i \geq u_i^+, i = 1, 2, \\ \tilde{u}_i, & \text{иначе} \end{cases}$$

где

$$\tilde{u}_1 = q_1(x^f - x) + q_2(y^f - y),$$

$$\tilde{u}_2 = q_3(\theta^f - \theta),$$

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

В принятых для генетического программирования обозначениях с учетом номеров элементов во множествах (3.28), (2.46)-(2.48) запись базисного решения имеет вид

$$S_1 = \left(\begin{bmatrix} 2 \\ 0 \end{bmatrix}, \begin{bmatrix} 2 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 3 \end{bmatrix}, \begin{bmatrix} 2 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 4 \end{bmatrix} \right),$$

$$S_2 = \left(\begin{bmatrix} 2 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 2 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 5 \end{bmatrix} \right).$$

Для учета ограничения используем функцию штрафа, которая добавляем к значению функционала (3.26).

$$p_i = \begin{cases} 0, & \text{если } \Delta x \Delta y < 0 \\ \int_0^{t_f} \left(\sqrt{(\Delta x)^2 + (\Delta y)^2} \right) dt - \text{иначе} & , i = \overline{1,4}, \end{cases} \quad (3.29)$$

где

$$\Delta x = \min_j \left\{ \left(x^{\alpha_j} - x_i \right), \left(x^{\beta_j} - x_i \right) \right\}, \quad i = \overline{1,4}, \quad (3.30)$$

$$\Delta y = \min_j \left\{ \left(y^{\alpha_j} - y_i \right), \left(y^{\beta_j} - y_i \right) \right\}, \quad i = \overline{1,4}. \quad (3.31)$$

С учетом множества точек начальных значений (3.27) и добавление критерия, определяющего попадание в терминальное состояние, критерии качества для задачи численного многокритериального структурно-параметрического синтеза системы управления мобильного робота имеют следующий вид:

$$\tilde{J}_1 = \sum_{j=1}^4 \left(t_f^{(j)} + \sum_{i=1}^4 p_i \right)_{x_{0,j}, y_{0,j}, \theta_{0,j}} \rightarrow \min, \quad (3.32)$$

$$\tilde{J}_2 = \sum_{j=1}^4 \left(x^2(t_f^{(j)}) + y^2(t_f^{(j)}) + \theta^2(t_f^{(j)}) + \sum_{i=1}^4 p_i \right)_{x_{0,j}, y_{0,j}, \theta_{0,j}} \rightarrow \min, \quad (3.33)$$

где

$$t_f^{(j)} = \begin{cases} t, & \text{если } \left(x^2(t) + y^2(t) + \theta^2(t) \right)_{x_{0,j}, y_{0,j}, \theta_{0,j}} \leq \varepsilon \\ t^+, & \text{если } \left(x^2(t^+) + y^2(t^+) + \theta^2(t^+) \right)_{x_{0,j}, y_{0,j}, \theta_{0,j}} > \varepsilon \end{cases}, \quad j = \overline{1,4}, \quad (3.34)$$

t^+ - заданная положительная величина.

В соотношениях (3.32) – (3.34) нижний индекс $(\dots)_{x_{0,j}, y_{0,j}, \theta_{0,j}}$ указывает на то, что решения $x(t)$, $y(t)$, $\theta(t)$ дифференциальных уравнений (3.1) – (3.3) получены при начальных значениях $x(0) = x_{0,j}$, $y(0) = y_{0,j}$, $\theta(0) = \theta_{0,j}$, $1 \leq j \leq 4$.

При расчетах были использованы параметры генетического алгоритма, приведенные в таблице 3.1.

Таблица 3.1 Параметры генетического алгоритма

Мощность множества возможных решений	$H = 1024$
Число поколений	$G = 256$
Число попыток скрещиваний в одном поколении	$R = 128$
Число вариаций одного возможного решения	$l = 8$
Максимальная длина записи кода для одного математического выражения	$ S = 48$
Число бит для кода целой части параметра	$c = 4$
Число бит для кода дробной части параметра	$d = 12$
Число поколений между сменой базисного решения	$Epo = 24$
Вероятность мутации	$p_\mu = 0,7$
Параметр для определения вероятности скрещивания	$\gamma = 0,4$

Предельное время моделирования процесса управления составляло $t^+ = 15$

с.

При смене базисного решения из множества Парето выбиралось возможное решение по критерию близости к утопической точке в пространстве нормированных функционалов. Для выбора номера возможного решения через каждые E_{pro} поколений первоначально нормируем все значения функционалов для всех возможных решений $(\mathbf{W}_i, \mathbf{y}^i)$, $i = \overline{0, H}$,

$$\bar{J}_k = \frac{J_k(i) - J_k^+}{J_k^+ - J_k^-}, \quad i = \overline{0, H}, \quad k = 1, 2, \quad (3.35)$$

где $J_k(i)$ значение функционала J_k , $k \in \{1, 2\}$, для решения $(\mathbf{W}_i, \mathbf{y}^i)$, $i \in \{0, \dots, H\}$, J_k^- , J_k^+ - соответственно, минимальное и максимальное значения функционала J_k , $k \in \{1, 2\}$.

Далее находим решение с наименьшей нормой векторного функционала $\bar{\mathbf{J}}(i) = [\bar{J}_1(i) \ \bar{J}_2(i)]^T$, $i \in \{0, \dots, H\}$.

$$\|\bar{\mathbf{J}}(i_-)\| = \min \left\{ \|\bar{\mathbf{J}}(i)\| : i = \overline{1, H} \right\}, \quad (3.36)$$

где

$$\|\bar{\mathbf{J}}(i)\| = \sqrt{J_1^2(i) + J_2^2(i)}. \quad (3.37)$$

Возможное решение $(\mathbf{W}_{i_-}, \mathbf{y}^{i_-})$ делаем новым базисным решением.

Вычисления выполнялись на компьютере с процессором Intel(R) Core(TM) i7-2640M CPU @ 2.80GHz 2.80 GHz. Время вычисления составляло около 45 мин.

В результате синтеза было получено следующее решение

$$\tilde{u}_1 = (x^f - x) - q_1^3 (x^f - x)$$

$$\tilde{u}_2 = \operatorname{sgn}(A^3) \left(1 - e^{-|A^3|} \right),$$

$$A = \min \left\{ \arctan(y^f - y), q_2^2 \right\} + \operatorname{sgn}(B) \left(e^{-|B|} - 1 \right) (x^f - x),$$

$$B = \arctan \left(\sin \left(\frac{1 - e^{-(\theta^f - \theta)}}{1 + e^{-(\theta^f - \theta)}} \right) - \sin^3 \left(\frac{1 - e^{-(\theta^f - \theta)}}{1 + e^{-(\theta^f - \theta)}} \right) \right),$$

$$q_1 = 11,80249, \quad q_2 = 14,19629.$$

На рисунок 3.2-3.14 приведены результаты моделирования полученной системы управления мобильным роботом.

Согласно траекториям движения центра масс, представленных на рисунок 3.2, мобильный робот достигает терминального состояния из всех начальных значений. Траектории движения углов робота (см. рис 3.3-3.6) показывают, что пространственные ограничения выполняются.

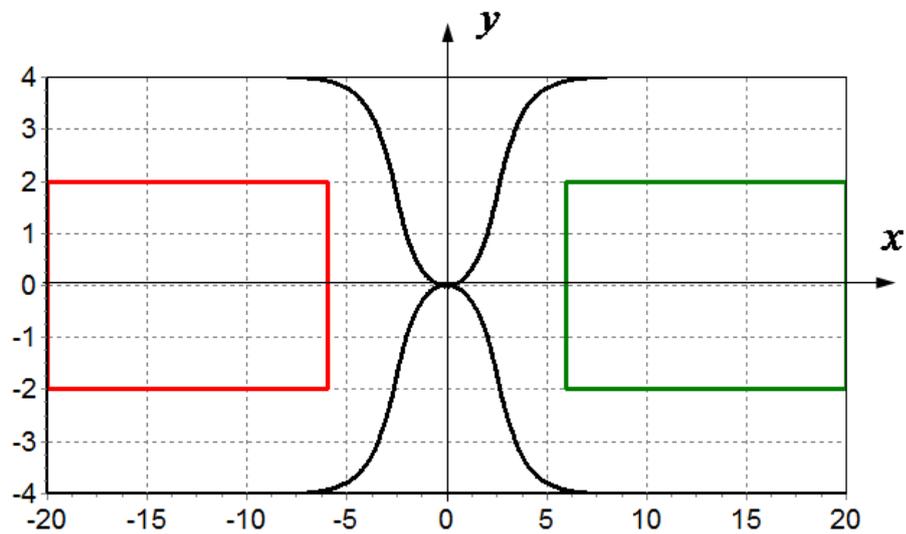


Рисунок 3.2 Траектории движения центра масс мобильного робота

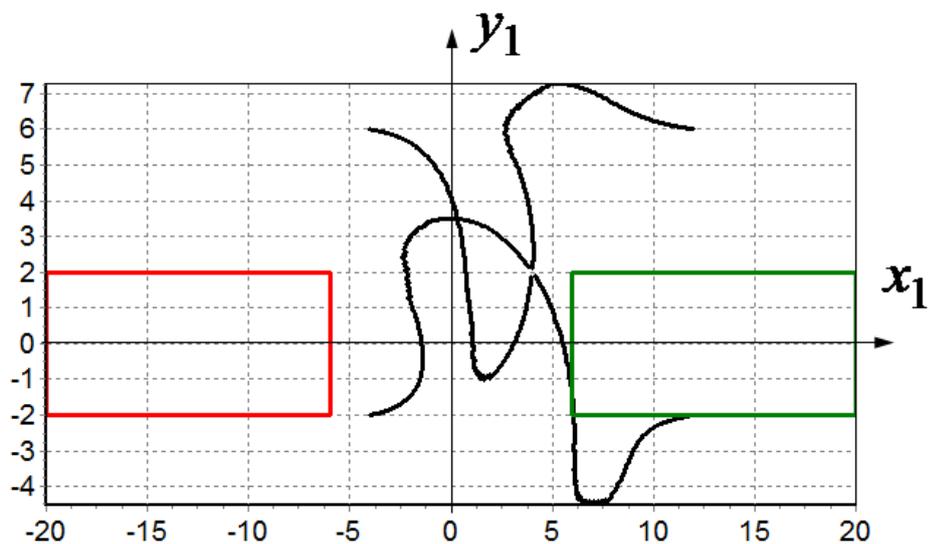


Рисунок 3.3 Траектории движения левого переднего угла мобильного робота

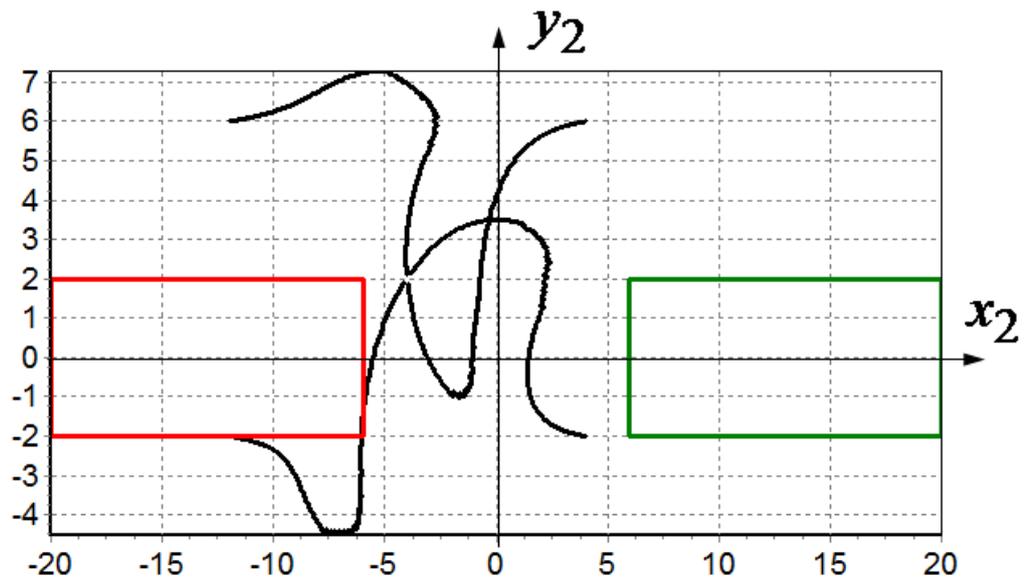


Рисунок 3.4 Траектории движения левого заднего угла мобильного робота

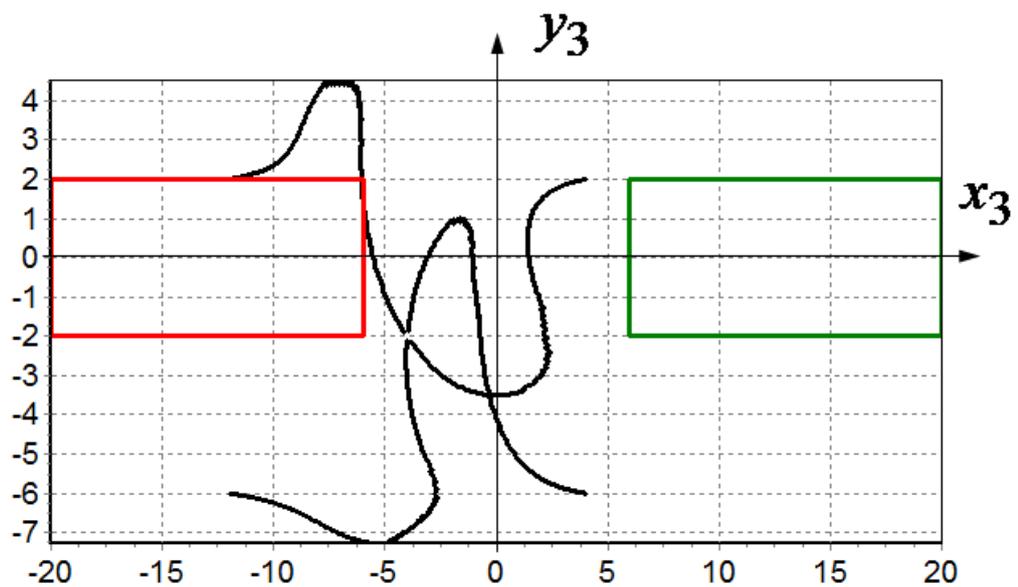


Рисунок 3.5 Траектории движения правого заднего угла мобильного робота

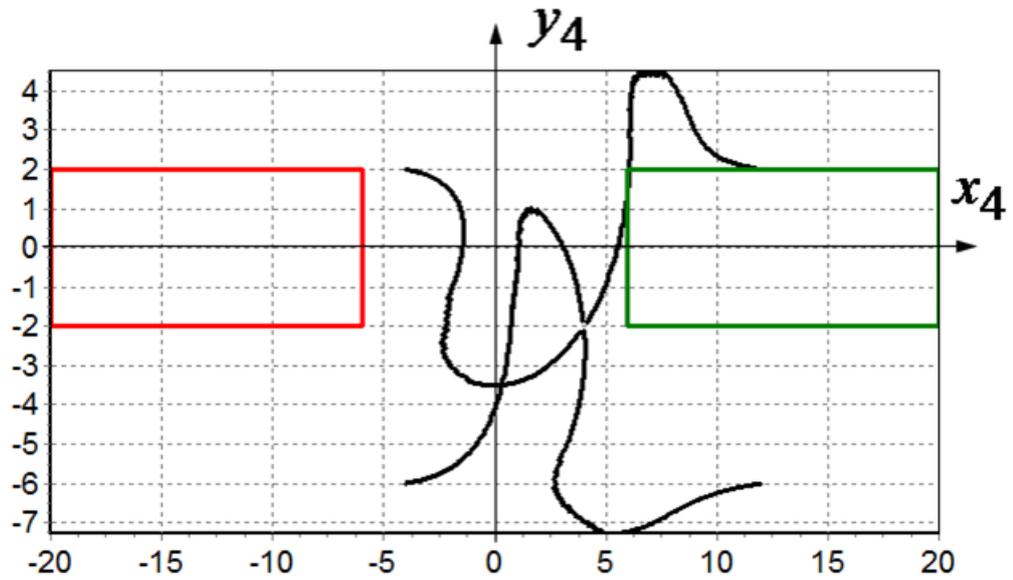


Рисунок 3.6 Траектории движения правого переднего угла мобильного робота

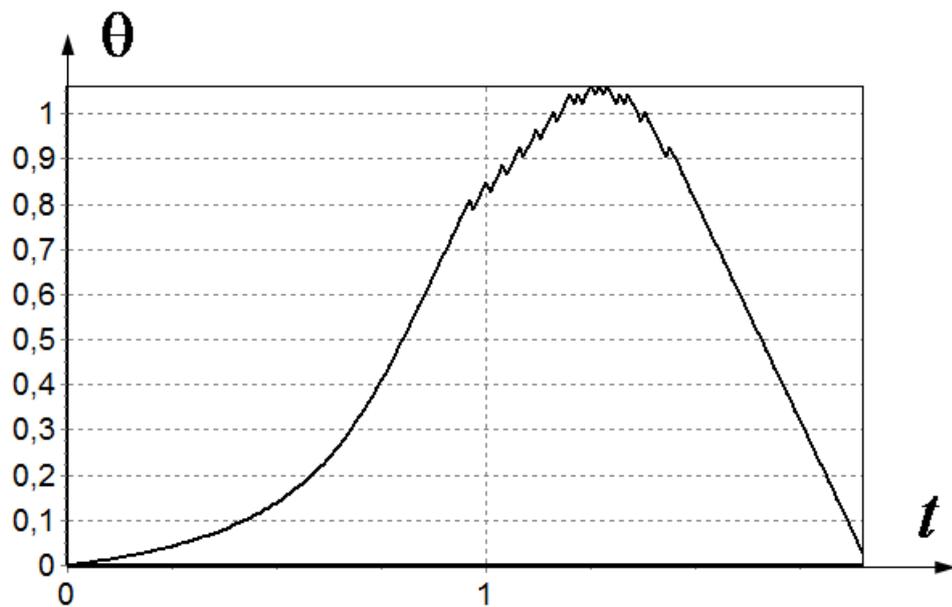


Рисунок 3.7 Изменение угла направления скорости при начальных значениях
 $x(0) = -8$, $y(0) = -4$, $\theta(0) = 0$

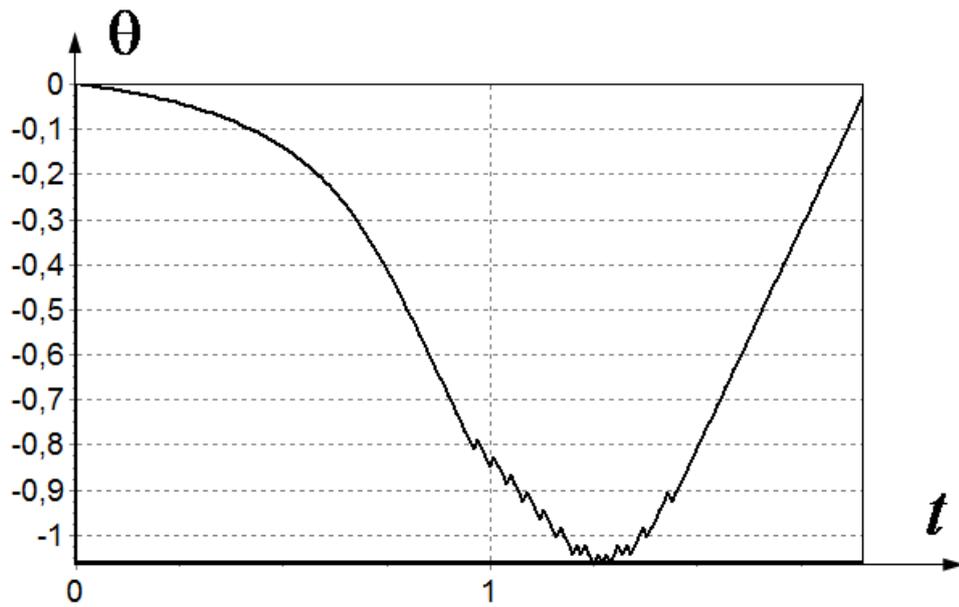


Рисунок 3.8 Изменение угла направления скорости при начальных значениях

$$x(0) = 8, \quad y(0) = -4, \quad \theta(0) = 0$$

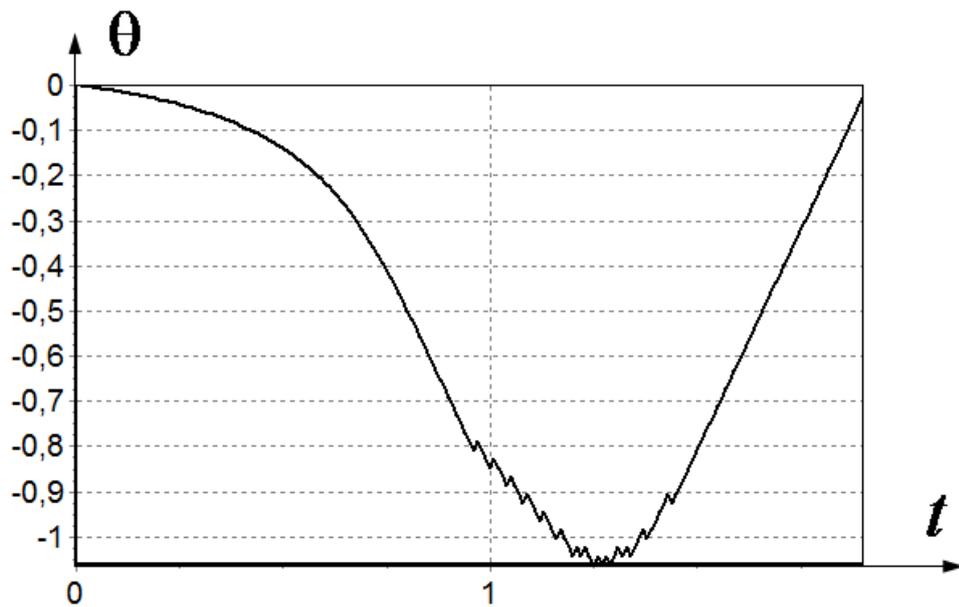


Рисунок 3.9 Изменение угла направления скорости при начальных значениях

$$x(0) = -8, \quad y(0) = 4, \quad \theta(0) = 0$$

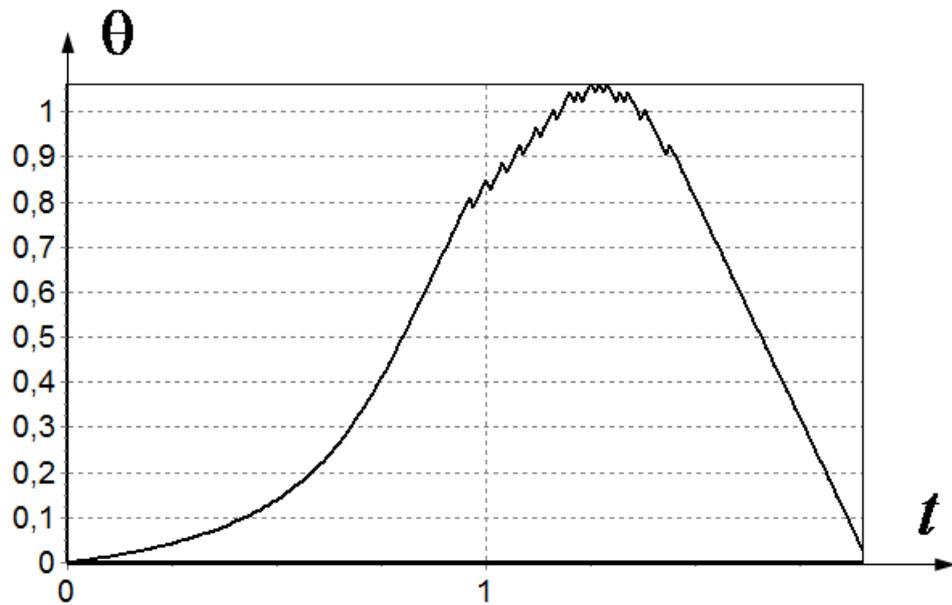


Рисунок 3.10 Изменение угла направления скорости при начальных значениях

$$x(0)=8, y(0)=4, \theta(0)=0$$

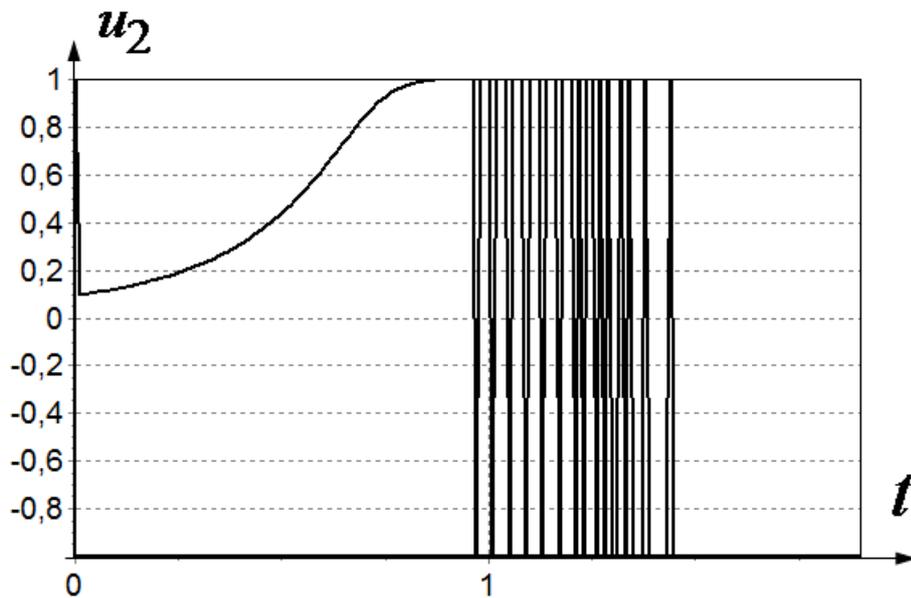


Рисунок 3.11 Изменение второго компонента управления при начальных

значениях

$$x(0)=-8, y(0)=-4, \theta(0)=0$$

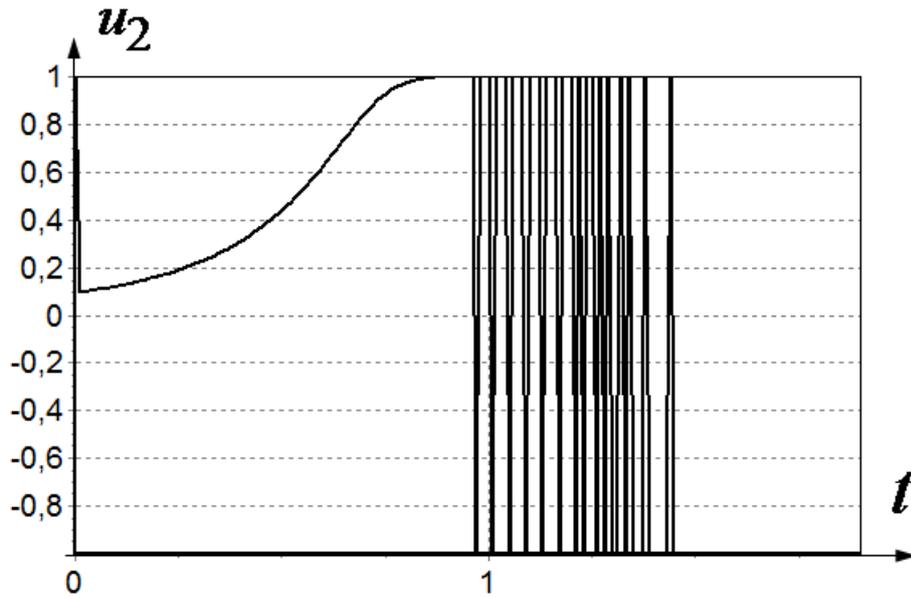


Рисунок 3.12 Изменение второго компонента управления при начальных значениях

$$x(0) = 8, \quad y(0) = -4, \quad \theta(0) = 0$$

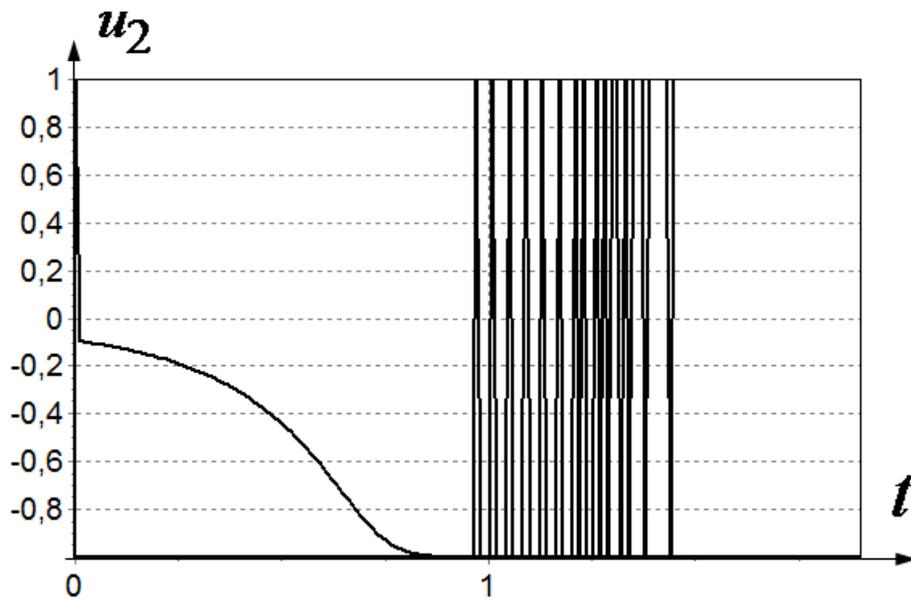


Рисунок 3.13 Изменение второго компонента управления при начальных значениях

$$x(0) = -8, \quad y(0) = 4, \quad \theta(0) = 0$$

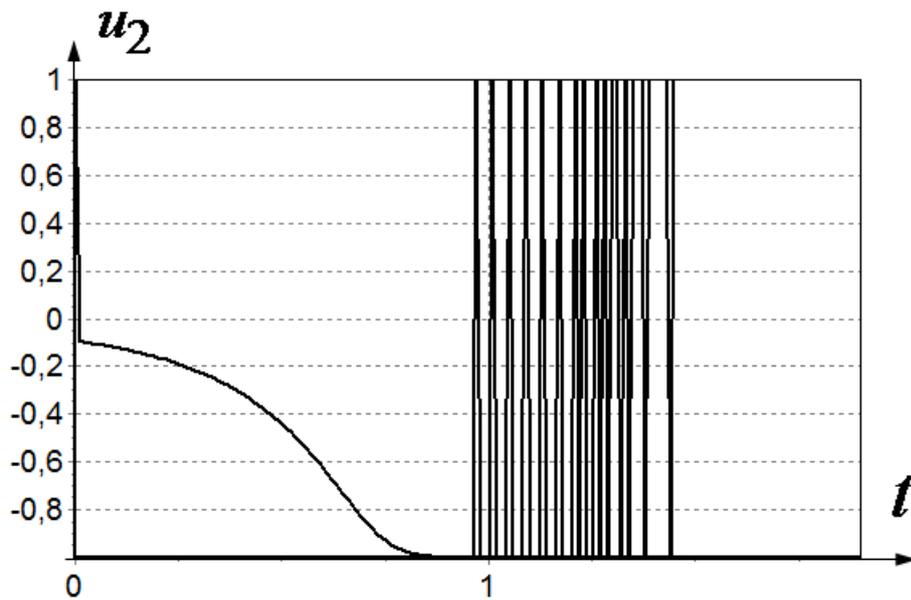


Рисунок 3.14 Изменение второго компонента управления при начальных значениях

$$x(0) = 8, y(0) = 4, \theta(0) = 0$$

3.2 Синтез системы управления движением мобильного робота по пространственным траекториям

Данная задача требует нахождения математического выражения функции, описывающей зависимость значений вектора управления от вектора состояния объекта управления. Трудность решения задачи синтеза заключается в том, что помимо минимизации критерия качества, заданного в виде интегрального функционала, найденная функция управления должна еще обеспечивать и достижение объектом цели управления. При эволюционном поиске в начальном множестве случайно сгенерированных возможных решений очень небольшая часть решений обеспечивает достижение цели управления. Это обстоятельство приводит к значительному увеличению времени поиска решения. Согласно данному принципу исследователь задает первоначально базисное решение, для задач синтеза систем управления это решение, в первую очередь, должно обеспечить достижение цели управления. Начальное множество возможных

решений генерируется как множество наборов вариаций базисного решения. Эволюционный алгоритм поиска выполняет все необходимые операции, в частности, если используется генетический алгоритм, то операции скрещивания и мутации, на наборах малых вариаций базисного решения. В процессе поиска оптимального решения базисное решение заменяется на наилучшее найденное к этому моменту решение.

Метод вариационного генетического программирования сравнивался с методом генетического программирования [2] и показал свою эффективность при решении задачи синтеза оптимального управления. Методом вариационного генетического программирования решено несколько задач синтеза управления, в частности, задача оптимальной парковки мобильного робота из различных начальных условий с учетом пространственных ограничений [1]. Настоящая работа посвящена продолжениям исследований метода вариационного генетического программирования для синтеза систем управления. В работе рассматривается популярная для мобильных роботов задача синтеза оптимального управления движением вдоль пространственной траектории. Особенностью задачи является необходимость получения одной системы управления, обеспечивающей точное движение робота по различным пространственным траекториям.

3.2.1 Задача синтеза управления движением вдоль пространственной траектории

Рассмотрим формальную постановку задачи синтеза управления траекторным движением. Задана математическая модель объекта управления.

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}), \quad (3.38)$$

где \mathbf{x} - вектор состояния объекта, \mathbf{u} - вектор управления, $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{u} \in U \subseteq \mathbb{R}^m$, U - замкнутое ограниченное множество, $m \leq n$.

Задана область начальных условий

$$\mathbf{x}(0) \in X_0 \subseteq \mathbb{R}^n$$

Заданы k пространственных траекторий движения в форме $n-r$ -мерных многообразий

$$\eta_{i,j}(\mathbf{x}) = 0, i = \overline{1, r}, j = \overline{1, k} \quad (3.39)$$

Траектории (3.39) называем пространственными, так как они не зависят от времени.

Вместе с траекториями (3.39) заданы также начальная и конечная точки

$$\mathbf{x}^{0,j}, \mathbf{x}^{f,j}, \eta_{i,j}(\mathbf{x}^{0,j}) = 0, \eta_{i,j}(\mathbf{x}^{f,j}) = 0, i = \overline{1, r}, j = \overline{1, k}. \quad (3.40)$$

Необходимо найти управление в форме

$$\mathbf{u} = \mathbf{h}(\mathbf{x}), \quad (3.41)$$

где $\mathbf{h}(\mathbf{x})$ - искомая синтезирующая функция, $\mathbf{h}(\mathbf{x}): \mathbb{R}^n \rightarrow \mathbb{R}^m$, удовлетворяющая следующим свойствам:

- $\forall \mathbf{x} \in \mathbb{R}^n, \mathbf{h}(\mathbf{x}) \in U$;
- $\forall \mathbf{x}(0) = \mathbf{x}^0 \in X_0 \subseteq \mathbb{R}^n$ решение $\mathbf{x}(t, \mathbf{x}^0)$ системы $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{h}(\mathbf{x}))$

должно обеспечивать минимум функционалу и удовлетворять условиям попадания в начальную и конечную точки траектории

$$J(\mathbf{x}(t, \mathbf{x}^0)) = \sum_{j=1}^k \sqrt{\sum_{i=1}^r \int_0^{t_{f,j}} \eta_{i,j}^2(\mathbf{x}(t, \mathbf{x}^0))} \rightarrow \min, \quad (3.42)$$

$$\mathbf{x}(t_{0,j}, \mathbf{x}^0) = \mathbf{x}^{0,j}, \mathbf{x}(t_{f,j}, \mathbf{x}^0) = \mathbf{x}^{f,j}, \quad 0 \leq t_{0,j} < t_{f,j} < \infty. \quad (3.43)$$

В качестве технической стратегии реализации системы управления, обеспечивающей движение по траектории, зададим точки на каждой заданной траектории

$$T_j = (\mathbf{x}^{0,j}, \mathbf{x}^{1,j}, \dots, \mathbf{x}^{p_j,j} = \mathbf{x}^{f,j}), \quad j = \overline{1, k}. \quad (3.44)$$

где

$$\forall \mathbf{x}^{v,j} \in T_j \quad \eta_{i,j}(\mathbf{x}^{v,j}) = 0, \quad j = \overline{1, k}, \quad (3.45)$$

$$\|\mathbf{x}^{v,j} - \mathbf{x}^{v-1,j}\|_2 \leq d_j, \quad v = \overline{1, p}, \quad j = \overline{1, k}, \quad (3.46)$$

d_j - заданная величина, определяющая количество точек p_j на траектории Γ_j , $j = \overline{1, k}$.

Система управления движением объекта по траектории представляет собой систему стабилизации объекта относительно заданной точки из множества (3.44) с дополнительным условием переключения точек.

Для системы стабилизации относительно пространственной точки синтезирующая функция (3.41) в качестве аргумента должна использовать отклонение от заданной целевой точки

$$\mathbf{u} = \mathbf{h}(\mathbf{x}^{v,j} - \mathbf{x}(t, \mathbf{x}^0)), \quad (3.47)$$

В данном случае система управления должна дополнительно включать подсистему переключения целевых точек, например, по условию ее достижения или по истечении некоторого заданного промежутка времени

$$v \leftarrow v + 1, \text{ если } (\|\mathbf{x}^{v,j} - \mathbf{x}(t, \mathbf{x}^0)\|_2 \leq \varepsilon) \vee (t - t_{v-1} < \Delta t), \quad (3.48)$$

где ε и Δt - заданные положительные величины, определяющие точность и скорость движения по траектории.

Для решения задачи (3.38-3.43) синтеза системы управления движением объекта по пространственной траектории со стратегией (3.44)-(3.48) стабилизации относительно точек траектории используем метод вариационного генетического программирования [1], [3].

3.2.2 Пример синтеза системы управления движением по траектории

В качестве примера рассмотрим задачу синтеза системы управления движением мобильного робота по заданным пространственным траекториям.

Математическая модель объекта управления имеет следующий вид

$$\dot{x} = u_1 \cos(\theta), \quad \dot{y} = u_1 \sin(\theta), \quad \dot{\theta} = \frac{u_2}{L} \tan(u_2), \quad (3.49)$$

где L - постоянный параметр объекта.

Управление объектом ограничено в виде

$$u_1^- \leq u_1 \leq u_1^+, \quad u_2^- \leq u_2 \leq u_2^+. \quad (3.50)$$

Для системы (3.49) задано множество начальных условий

$$X_0 = \left\{ \begin{bmatrix} x_{0,1} & y_{0,1} & \theta_{0,1} \end{bmatrix}^T, \dots, \begin{bmatrix} x_{0,M} & y_{0,M} & \theta_{0,M} \end{bmatrix}^T \right\}. \quad (3.51)$$

Заданы пространственные траектории в виде

$$\eta_j(x, y) = 0, \quad j = \overline{1, k}. \quad (3.52)$$

В качестве траекторий рассматривались $k = 4$ траектории, две из которых были гладкими, а две – с разрывами по производным.

Синтез системы управления выполнялся методом вариационного генетического программирования. Рассматривалась стратегия управления стабилизации точек траектории. При вычислительном эксперименте были приняты следующие значения параметров модели и задачи: $L = 4$, $u_1^- = -5$, $u_1^+ = 5$, $u_2^- = -\pi/2$, $u_2^+ = \pi/2$, $M = 4$, $x_{0,1} = -0,5$, $y_{0,1} = -0,25$, $\theta_{0,1} = 0$, $x_{0,2} = -0,25$, $y_{0,2} = -0,25$, $\theta_{0,2} = 0$, $x_{0,3} = -0,5$, $y_{0,3} = 0,25$, $\theta_{0,3} = 0$, $x_{0,4} = -0,25$, $y_{0,4} = 0,25$, $\theta_{0,4} = 0$, $\varepsilon = 0,01$, $\Delta t = 2$ с., $d = 0,32$, $p = 32$.

В результате синтеза была получена следующая синтезирующая функция:

$$u_i = \begin{cases} u_i^-, & \text{если } \tilde{u}_i < u_i^- \\ u_i^+, & \text{если } \tilde{u}_i > u_i^-, \\ \tilde{u}_i, & \text{иначе} \end{cases} \quad (3.53)$$

где

$$\tilde{u}_1 = \arctan(f_{3,3}(q_1, \delta, \operatorname{sgn}(A)(e^{|A|} - 1)) - f_{3,3}^3(q_1, \delta, \operatorname{sgn}(A)(e^{|A|} - 1))),$$

$$\tilde{u}_2 = f_{3,1}(\operatorname{sgn}(\sin(\delta))e^{-|\sin(\delta)|}, \sin^3(e^{q_1} - e^{3q_1}), q_2),$$

$$\delta = \theta - \arctan\left(\frac{y^{v,j} - y}{x^{v,j} - x}\right), \quad A = \mu\left(\sqrt{\sin^2(q_2) + \ln^2|\sin(\delta)|}\right), \quad \mu(\alpha) = \begin{cases} \alpha, & \text{если } |\alpha| < 1 \\ \operatorname{sgn}(\alpha), & \text{иначе} \end{cases},$$

$$f_{3,3}(\alpha, \beta, \gamma) = \begin{cases} \beta - \alpha, & \text{если } |\alpha - \beta| < |\alpha - \gamma| \\ \gamma - \alpha, & \text{иначе} \end{cases}, \quad f_{3,1}(\alpha, \beta, \gamma) = \begin{cases} \gamma, & \text{если } \alpha < \beta \\ -\gamma, & \text{иначе} \end{cases},$$

$$q_1 = 11,29565, \quad q_2 = 12,13770.$$

На рис 1-4 приведены графики движения мобильного робота с полученным управлением по заданным пространственным траекториям. Там же приведены целевые точки траекторий, помеченные маркерами.

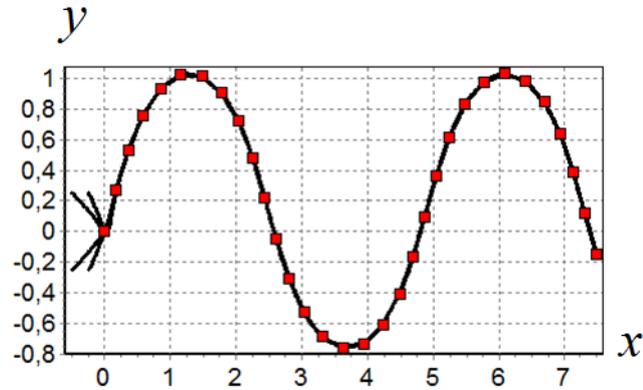


Рисунок 3.15 Движение робота по траектории 1

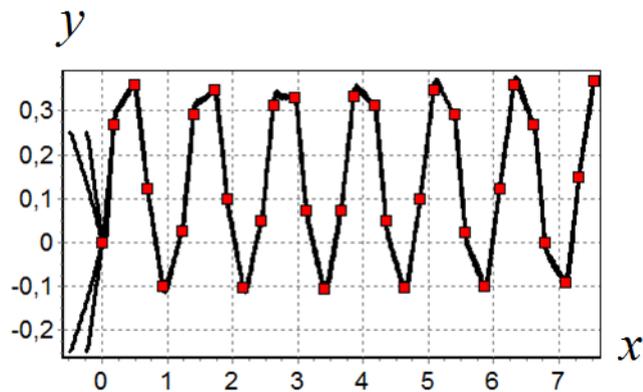


Рисунок 3.16 Движение робота по траектории 2

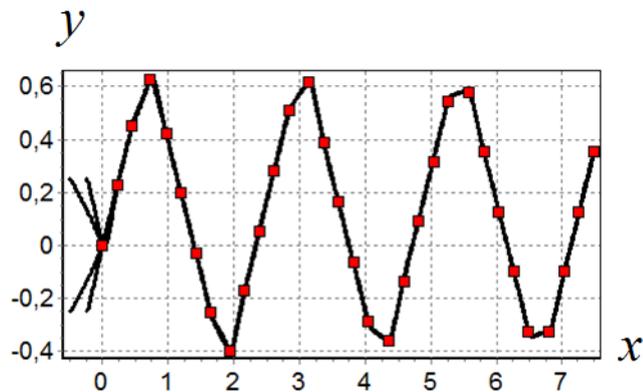


Рисунок 3.17 Движение робота по траектории 3

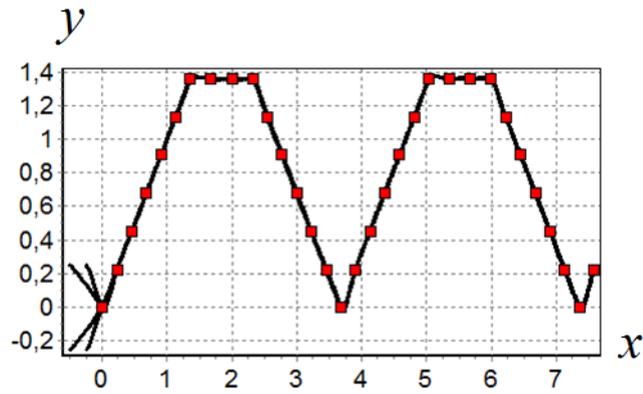


Рисунок 3.18 Движение робота по траектории 4

Как видно из результатов моделирования полученная нелинейная система управления обеспечивает достаточно качественное движение робота по заданным пространственным траекториям. Точность прохождения целевых точек траектории достаточно высока. Начальные условия практически не влияют на качество управления.

Работа выполнена по гранту РФФИ № 14-08-00008-а.

3.3 Выводы к главе 3

Решена задача синтеза системы управления мобильным роботом методом вариационного генетического программирования. Полученная в результате синтеза система управления является нелинейной и обеспечивает достижение мобильным роботом терминального состояния из всех заданных начальных значений. Моделирование показало, что траектории движения углов мобильного в процессе управления не нарушают пространственных фазовых ограничений.

ЗАКЛЮЧЕНИЕ

1. Разработан метод вариационного генетического программирования для решения задачи синтеза системы управления. Метод включает основные преимущества известных методов. Метод вариационного генетического программирования использует принцип малых вариаций базисного решения, заимствованный из метода сетевого оператора и представление математического выражение в форме строки кодов элементов математического выражения.
2. Разработан метод кодирования математических выражений в форме числовых двухкомпонентных векторов. Первый компонент вектора указывает на число аргументов кодируемой функции. Второй компонент вектора указывает на номер функции из заданного множества функций.
3. Разработан метод поиска оптимального решения для задачи синтеза системы на основе принципа малых вариаций базисного решения. Определены малые вариации кода записи математического выражения. Метод вариационного генетического программирования использует четыре малых вариации кода.
4. Решена задача синтеза системы управления мобильным роботом методом вариационного генетического программирования. Полученная в результате синтеза система управления является нелинейной и обеспечивает достижение мобильным роботом терминального состояния из всех заданных начальных значений. Моделирование показало, что траектории движения углов мобильного в процессе управления не нарушают пространственных фазовых ограничений.
5. Разработан программный комплекс, реализующий вычислительный метод для решения задачи синтеза системы управления мобильным роботом четырех колесного типа на основе вариационного

генетического программирования. Программный комплекс апробирован на решении практических задач.

СПИСОК СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ

АКОР – аналитического конструирования оптимальных регуляторов

АКАР – аналитического конструирования агрегированных регуляторов

ПИД-регулятор – пропорционально интегрально дифференцирующий регулятор

PSO – метод роя частиц

ГА – генетический алгоритм

МГП – метод генетического программирования

МГЭ – метод грамматической эволюции

МАП – метод аналитического программирования

МСО – метод сетевого оператора

МВГП – метод вариационного генетического программирования

МВАП – метод вариационного аналитического программирования

МВБР – метод вариаций базисного решения

Литература

1. Алексеев, В.М. Оптимальное управление / В.М. Алексеев, В.М. Тихомиров, С.В. Фомин – М.: Наука, 1979. – 384 с.
2. Антомонов, Ю.Г. Синтез оптимальных систем / Ю.Г. Антомонов – М.: Наукова думка, 1972. – 320 с.
3. Афанасьев, В.Н. Оптимальные системы управления / В.Н. Афанасьев – М.: Изд-во РУДН, 2007. – 260 с.
4. Афанасьев, В.Н. Управление неопределенными динамическими системами / В.Н. Афанасьев – М.: ФИЗМАТЛИТ, 2008. – 208 с.
5. Болтянский, В.Г. Математические методы оптимального управления / В.Г. Болтянский – М.: Наука, 1968. – 408 с.
6. Гилимьянов, Р.Ф. Управление движением колесного робота в задаче следования вдоль криволинейного пути. / Р.Ф. Гилимьянов, А.В. Пестерев, Л.Б. Рапопорт // Известия РАН. Теория и системы управления. - 2008. - № 6. - С. 209–216.
7. Дивеев, А.И. Метод сетевого оператора / А.И. Дивеев. - М.: ВЦ РАН, 2010. – 178 с.
8. Дивеев, А.И. Метод сетевого оператора и его применение в задачах управления / А.И. Дивеев, Е.А. Софронова. - М.: РУДН, 2012. – 182 с.
9. 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.
10. 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.
11. Дивеев, А.И. Решение задачи синтеза системы управления методом вариационного генетического программирования/ А.И. Дивеев, С.И. Ибадулла, Е.А. Софронова // Современные проблемы науки и образования. – 2013. – № 6; URL: <http://www.science-education.ru/113-11697>.

12. Дивеев, А.И. Сравнение методов генетического и вариационного генетического программирования на примере задачи синтеза управления для модели «Хищник-жертва»/ А.И. Дивеев, С.И. Ибадулла // Эл. научно-техн. изд. Наука и образование. Эл. № ФС 77 48211 Гос. рег. № 0421200025 ISSN 1994-0408 #5 май, 2014 г. DOI 10.7463/ 0514.0709252.
13. Дивеев, А.И. Численный метод вариационного генетического программирования для синтеза системы управления мобильного робота/ А.И. Дивеев, С.И. Ибадулла // Труды Международного симпозиума Надежность и качество ; под ред. Н.К. Юркова. – Пенза - 26 мая – 01 июня. - 2014. - Т. 1. - С. 30-35.
14. Дивеев, А.И. Метод вариационного генетического программирования для синтеза систем управления. / А.И. Дивеев, С.И. Ибадулла // Труды одиннадцатого международного симпозиума Интеллектуальные системы INTELS'2014 под ред. К.А. Пупкова. – Москва. - 30 июня-04 июля. - 2014. - С. 74-77.
15. 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.
16. Дивеев, А.И. Синтез системы управления методом сетевого оператора на основе аппроксимации множества оптимальных траекторий. / А.И. Дивеев, К.А. Пупков, Е.А. Софронова, Е.Ю. Шмалько // Труды XII Всероссийского совещания по проблемам управления ВСПУ-2014 Москва 16-19 июня 2014. - С. 8023-8033.
17. Дорф, Р. Современные системы управления / Р. Дорф, Р. Бишоп - М.: «Лаборатория базовых знаний», 2002. - 832 с.
18. Зубов, В.Н. Лекции по теории управления / В.Н. Зубов – СПб.: Издательство «Лань». – 2009. – 496 с.

19. 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.
20. 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.
21. 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.
22. 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.
23. Колесников, А.А. Синергетические методы управления сложными системами. Теория системного анализа / А.А. Колесников – М.: КомКнига, 2006. – 240 с.
24. Кондратьев, Г.В. Геометрическая теория синтеза оптимальных стационарных гладких систем управления / Г.В. Кондратьев – М.: ФИЗМАТЛИТ, 2003. – 144 с.
25. Крассовский, А.А. Системы управления полетом и их аналитическое конструирование / А.А. Крассовский. – М.: Наука, 1973. – 558 с.
26. Крутько, П.Д. Обратные задачи динамики управляемых систем. Нелинейные модели / П.Д. Крутько. - М.: Наука, 1987. - 304 с.
27. Кунцевич, В.М. Синтез систем автоматического управления с помощью функций Ляпунова. / В.М. Кунцевич, М.М. Лычак - М.: Наука, 1977. – 400 с.

28. Летов, А.М. Математическая теория процессов управления / А.М. Летов. – М.: Наука, 1981. – 255 с.
29. Ли, Э.Б. Основы теории оптимального управления. / Э.Б. Ли, Л. Маркус. – М.: Наука, 1972. – 576 с.
30. Мерриэм К. Теория оптимизации и расчет систем управления с обратной связью / Мерриэм. – М.: Мир, 1967. – 548 с.
31. 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.
32. 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.
33. Пестерев А.В. Стабилизация движения колесного робота вдоль криволинейной траектории, проложенной по неровной поверхности / А.В. Пестерев, Л.Б. Рапопорт // Известия РАН. Теория и системы управления. – 2010. - № 4. - С. 167-176.
34. Пестерев, А.В. Синтез стабилизирующего управления в задаче следования колесного робота вдоль заданной кривой / А.В. Пестерев // Автоматика и Телемеханика. 2012. - № 7. С. 25–39.
35. Пестерев, А. В. Каноническое представление задачи путевой стабилизации для колесных роботов. / А.В. Пестерев, Л.Б. Рапопорт // Автоматика и Телемеханика. 2013. - № 5. С. 80 - 101.
36. 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.
37. 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.

38. 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.
39. 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.
40. Ройтенберг, Я.Н. Автоматическое управление / Я.Н. Ройтенберг – М.: Наука, 1978. – 552 с.
41. Сейдж Э.П. Оптимальное управление системами. / Э.П. Сейдж, Ч.С. Уайт М.: Радио и связь, 1982.-392 с.
42. Синергетические методы управления сложными системами. Механические и электромеханические системы. / А.А. Колесников и др. – М.: КомКнига, 2006. - 304 с.
43. Справочник по теории автоматического управления; под ред. А.А. Красовского. - М.: Наука, 1987. – 712 с.
44. Atencia, V.J.M. The Network Operator Method for Synthesis of Intelligent Control System. / V.J.M. Atencia, 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.
45. 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.
46. Bourmistrova, A. Genetic Programming in Application to Flight Control System Design Optimisation / A. Bourmistrova, S. Khantsis. IN-TECH, 2010. P. 195 - 228.
47. 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.
48. 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.
49. Diveev, A.I. Application of network operator method for synthesis of optimal structure and parameters of automatic control system / A.I. Diveev, E.A. Sofronova // Proceedings of 17-th IFAC World Congress. – Seoul. - 2008, 05.07.2008 – 12.07.2008. - P. 6106 – 6113.
50. 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.
51. 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.
52. 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.
53. Koza, J. R. Genetic Programming: On the programming of computers by means of natural selection, Cambridge, Massachusetts.: MIT Press. -1992. – 819 p.
54. 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. - Vol. 1.- No 1. – P. 121 – 164.

ПРИЛОЖЕНИЯ

Для применения метода вариационного генетического программирования на математическую модель мобильного робота и проведения конкретного анализа и эксперимента была разработана специальная программа написанная на языке программирования Delphi 2007, профессором А.И. Дивеевым из Вычислительного центра им. А.А. Дородницына Российской академии наук и его аспирантами.

ПРОГРАММА

IMPLEMENTATION

```
//*****
Uses Functс,Unit6, Unit7, Unit8, Unit9, Unit10, TGAModelUnit,
  Unit3, Unit2, Unit14, Unit16, Unit17, Calc5, Calc4;
{$R *.dfm}
//*****
Procedure TForm1.BinaryOperations1Click(Sender: TObject);
Begin
  Form5:=TForm5.create(self);
  Form5.ShowModal;
End;
Procedure TForm1.Clear1Click(Sender: TObject);
Begin
  Memo1.Clear;
End;
//*****
Procedure TForm1.FormCreate(Sender: TObject);
var
  i,ii,j:integer;
Begin
  randomize;
  ProgressBar1.top:=212;
  ProgressBar1.left:=0;
  ProgressBar1.width:=ClientWidth;
  Memo1.Top:=0;
  Memo1.Left:=0;
  Memo1.Height:=progressbar1.top;
  Memo1.Width:=ClientWidth;
  Memo1.ReadOnly:=true;
  Setlength(x01,n1);
  Setlength(umin1,m1);
  Setlength(umax1,m1);
```

```

Setlength(q1,kR1);
Setlength(ym,ll1);
Setlength(um,m1);
Setlength(xf1,n1);
for i:=0 to nGraphc-1 do
begin
  SetLength(xmm[i],n1);
  SetLength(umm[i],m1);
end;
Setlength(qymax1,ny1);
Setlength(qymin1,ny1);
Setlength(stepsqy1,ny1);
for i:=0 to kR1-1 do
  q1[i]:=qc[i];
for i:=0 to n1-1 do
  x01[i]:=x0c[i];
for i:=0 to n1-1 do
  xf1[i]:=xfc[i];
for i:=0 to m1-1 do
  umin1[i]:=uminc[i];
for i:=0 to m1-1 do
  umax1[i]:=umaxc[i];
for i:= 0 to ny1-1 do
begin
  qymin1[i]:=qyminc[i];
  qymax1[i]:=qymaxc[i];
  stepsqy1[i]:=stepsqyc[i];
end;
SetLength(qyGraph,nGraphc,2);
End;
//*****
Procedure TForm1.GA1Click(Sender: TObject);
var
  i,j,k:integer;
Begin
  Form6:=TForm6.create(self);
  Form6.ShowModal;
  Form8:=TForm8.Create(self);
  Form8.ShowModal;
  klm:=Lmax1*Mout1;
  SetLength(yr1,klm);
  for i:=0 to klm-1 do
    for j:=0 to 1 do
      yr1[i,j]:=0;

```

```

for k := 0 to Mout1-1 do
  for i := 0 to high(y0Basc[k]) do
    for j := 0 to 1 do
      yr1[i+k*lmax1,j]:=y0Basc[k,i,j];
if Application.MessageBox('FOR YOUR INFORMATION: '
    +'When the object for GA with NOP is created then '
    +'you will not be able to change some paprameters. ',
    'Creation of object',
    MB_OKCANCEL+MB_ICONWARNING)=ID_OK then
begin
  EA:=TUser.Create(hh1, pp1, rr1, nfu1, lchr1, c1, d1, epo1,
    kel1, gamma1, pmut1, Lmax1, Mout1, kp1,
    kr1,n1,m1,ll1,ny1);
  EA.Setqymax(qymax1);
  EA.Setqymin(qymin1);
  EA.SetStepsqy(stepsqy1);
  GA1.Enabled:=false;
  Geneticalgorithm1.Enabled:=true;
  ParametersofGA1.Enabled:=true;
  EA.EndGeneration:=UpProgressBar;
end
else
  exit;
End;
//*****
Procedure TForm1.GAforNOPParameters1Click(Sender: TObject);
Begin
  ProgressBar1.Max:=EA.PP;
  EA.GenAlgorithm;
  ParetoSet1.Enabled:=true;
End;
//*****
Procedure TForm1.GAforParameters1Click(Sender: TObject);
Begin
  ProgressBar1.Max:=EA.PP;
  EA.GenAlgorithm1;
  ParetoSet1.Enabled:=true;
End;
//*****
Procedure TForm1.Geneticalgorithm1Click(Sender: TObject);
Begin
  Form7:=TForm7.create(self);
  Form7.ShowModal;
  Optimization1.Enabled:=true;

```

```

Expression1.Enabled:=true;
End;
//*****
Procedure TForm1.Model1Click(Sender: TObject);
var
  k,i:integer;
  tp:real;
Begin
  Form2:=TForm2.Create(self);
  Form2.ShowModal;
  EA.Initial;
  tp:=0;
  k:=0;
  EA.Setf0qset(EA.q);
  repeat
    if abs(EA.t-tp)<EA.dt/2 then
      begin
        k:=k+1;
        EA.Viewer;
        for i:=0 to ll1-1 do
          begin
            Setlength(ym[i],k);
            ym[i,k-1]:=EA.y[i];
          end;
        for i:=0 to m1-1 do
          begin
            Setlength(um[i],k);
            um[i,k-1]:=EA.u[i];
          end;
        Setlength(tm,k);
        tm[k-1]:=EA.t;
        tp:=tp+ntp;
      end;
    EA.Euler2;
  until (EA.t>tf1)or (Normdist(EA.x,xf1)<epsterm);
  f1:=EA.t;
  f0:=Normdist(EA.x,xf1);
  Form11:=TForm11.create(self);
  Form11.ShowModal;
End;

Procedure TForm1.ParametersofGA1Click(Sender: TObject);
Begin
  Form9:=TForm9.create(self);

```

```

    Form9.ShowModal;
End;
Procedure TForm1.Parametersofmodel1Click(Sender: TObject);
Begin
    Form3:=TForm3.Create(self);
    Form3.ShowModal;
End;
//*****
Procedure TForm1.Paretoset1Click(Sender: TObject);
var
    i:integer;
Begin
    Form10:=TForm10.create(self);
    Form10.ShowModal;
    label1.Caption:=inttostr(kchoose);
    EA.ReadChromosome(kchoose);
End;
//*****
Procedure TForm1.Pascal1Click(Sender: TObject);
var
    i:integer;
Begin
    EA.YrtoPasStr;
    for i:=0 to Mout1-1 do
        memo1.Lines.Add(EA.ystr[i]);
End;
//*****
Procedure TForm1.Savetofile1Click(Sender: TObject);
Begin
    if savedialog1.Execute then
        Memo1.Lines.SaveToFile(savedialog1.FileName);
End;
//*****
Procedure TForm1.SimMult1Click(Sender: TObject);
var
    k,i,j,iGraph:integer;
    tp,sumdelt,sumt:real;
Begin
    Form17:=TForm17.Create(self);
    Form17.ShowModal;
    sumt:=0;
    sumdelt:=0;
    for iGraph := 0 to nGraphc-1 do
        begin

```

```

for i:=0 to 1 do
  EA.qy[i]:=qyGraph[iGraph,i];
EA.Initial;
tp:=0;
k:=0;
EA.Setf0qSet(EA.q);
f0:=0;
repeat
  if abs(EA.t-tp)<EA.dt/2 then
  begin
    k:=k+1;
    for i:=0 to n1-1 do
    begin
      Setlength(xmm[iGraph,i],k);
      xmm[iGraph,i,k-1]:=EA.x[i];
    end;
    for i:=0 to m1-1 do
    begin
      Setlength(umm[iGraph,i],k);
      umm[iGraph,i,k-1]:=EA.u[i];
    end;
    if iGraph=0 then
    begin
      Setlength(tm,k);
      tm[k-1]:=EA.t;
    end;
    tp:=tp+ntp;
  end;
  EA.Euler2;
until (EA.t>tf1) or (Normdist(EA.x,xf1)<epsterm);
sumt:=sumt+EA.t;
sumdelt:=sumdelt+Normdist(EA.x,xf1);
end;
f0:=sumdelt;
f1:=sumt;
Form16:=TForm16.create(self);
Form16.ShowModal;
End;

//*****
Procedure TForm1.Unaryoperations1Click(Sender: TObject);
Begin
  Form4:=TForm4.create(self);
  Form4.ShowModal;

```

```

End;
//*****
Procedure TForm1.Undefinedparameters1Click(Sender: TObject);
var
  i:integer;
Begin
  Form14:=TForm14.Create(Self);
  Form14.ShowModal;
  EA.Setqymin(qymin1);
  EA.Setqymax(qymax1);
  for i:=0 to EA.ny-1 do
    EA.qy[i]:=EA.qymin[i];
  EA.Setstepsqy(stepsqy1);
  for i:=0 to ny1-1 do
    EA.ix[i]:=trunc((qymax1[i]-qymin1[i])/stepsqy1[i]);
  EA.Setixmax(EA.ix);
End;
//*****
Procedure UpProgressBar;
Begin
  Form1.ProgressBar1.SteptIt;
  Form1.Refresh;
End;
{ TGANOPUser }
//*****
Constructor TUser.Create(hh1, pp1, rr1, nfu1, lchr1, c1, d1, epo1,
  kel1:integer; gamma1, pmut1:real;
  Lmax1, Mout1, kp1,kr1,n1,m1,ll1,ny1:integer);
Begin
  Inherited Create(hh1, pp1, rr1, nfu1, lchr1, c1, d1, epo1,
    kel1, gamma1, pmut1,
    lmax1, Mout1, kp1, kr1,n1,m1,ll1,ny1);
End;
//*****
Procedure TUser.Func(var Fu: TArrReal);
var
  sumpen,shtraf,promah:real;
  i:integer;
Begin
  Initial;
  sumpen:=0;
  shtraf:=0;
  for i:=0 to kR-1 do
    f0qset[i]:=q[i];

```

```

repeat
  Euler2;
  Viewer;
  sumpen:=sumpen+check4(y);
until (t>tf1)or (Normdist(x,xf1)<epsterm);
fu[1]:=t+sumpen*dt;
promah:=Normdist(x,xf1);
fu[0]:=promah+sumpen*dt;
End;

```

```

Procedure TUser.Initial;

```

```

Begin

```

```

  x[0]:=qy[0];

```

```

  x[1]:=qy[1];

```

```

  x[2]:=x0[2];

```

```

  u[0]:=0;

```

```

  u[1]:=0;

```

```

  t:=0;

```

```

End;

```

```

//*****

```

```

Procedure TUser.RP(t1:real;x1:TArrReal;var f1:TArrReal);

```

```

var

```

```

  i:integer;

```

```

Begin

```

```

  f0xset[0]:=xf1[0]-x1[0];

```

```

  f0xset[1]:=xf1[1]-x1[1];

```

```

  f0xset[2]:=xf1[2]-x1[2];

```

```

  RPControl_GP;

```

```

  u[0]:=yout[0];

```

```

  u[1]:=yout[1];

```

```

  OgrUpr;

```

```

  f1[0]:=u[0]*cos(x1[2]);

```

```

  f1[1]:=u[0]*sin(x1[2]);

```

```

  f1[2]=(u[0]/Lbasc)*sin(u[1])/cos(u[1]);

```

```

  for i := 0 to n-1 do

```

```

    if abs(f1[i])>infinity then

```

```

      f1[i]:=Fa1_10(f1[i])*infinity;

```

```

End;

```

```

//*****

```

```

Procedure TUser.Viewer;

```

```

var

```

```

  i:integer;

```

```

  xlf,xlb,xrf,xrb,

```

```

  ylf,ylb,yrf,yrb:real;

```

```

    steta,cteta:real;
Begin
  y[0]:=x[0];
  y[1]:=x[1];
  y[2]:=x[2];
  steta:=sin(x[2]);
  cteta:=cos(x[2]);
  xlf:=x[0]*cteta+x[1]*steta+A1ac;
  ylf:=-x[0]*steta+x[1]*cteta+H1hc;

  xlb:=x[0]*cteta+x[1]*steta-B1bc;
  ylb:=ylf;

  xrf:=xlf;
  yrf:=-x[0]*steta+x[1]*cteta-H1hc;

  xrb:=xlb;
  yrb:=yrf;

  y[3]:=xlf*cteta-ylf*steta;
  y[4]:=xlf*steta+ylf*cteta;

  y[5]:=xlb*cteta-ylb*steta;
  y[6]:=xlb*steta+ylb*cteta;

  y[7]:=xrb*cteta-yrb*steta;
  y[8]:=xrb*steta+yrb*cteta;

  y[9]:=xrf*cteta-yrf*steta;
  y[10]:=xrf*steta+yrf*cteta;

End;
//*****
Function TermStop:boolean;
var
  i:integer;
  sum:real;
Begin
  sum:=0;
  for i:=0 to EA.n-1 do
    sum:=sum+sqr(EA.x[i]-xf1[i]);
  if sqrt(sum)<epsterm then
    result:=true
  else

```

```

    result:=false;
End;
//*****
Function Normdist(x1,xf1:TArrReal):real;
var
    sum:real;
    i:integer;
Begin
    sum:=0;
    for i:=0 to high(x1)do
        sum:=sum+sqr(xf1[i]-x1[i]);
    result:=sqrt(sum);
End;
//*****
Function Check4(y:TArrReal):real;
var
    i:integer;
    sum:real;
Begin
    sum:=0;
    for i := 0 to 3 do
        sum:=sum+Check(y[3+2*i],y[4+2*i]);
    result:=sum;
End;
Function Check(x,y:real):real;
var
    i:integer;
    dy,dx,dy1,dx1:real;
Begin
    i:=-1;
    repeat
        i:=i+1;
    until (i>1)or((x>= prepc[i,0,0]) and (x<=prepc[i,2,0])and
        (y>= prepc[i,0,1])and (y<=prepc[i,2,1]));
    if i<=1 then
    begin
        dy:=abs(prepc[i,0,1]-y);
        dy1:=abs(prepc[i,2,1]-y);
        dx:=abs(prepc[i,0,0]-x);
        dx1:=abs(prepc[i,2,0]-x);
        if dx1<dx then dx:=dx1;
        if dy1<dy then dy:=dy1;
        result:=sqrt(sqr(dx)+sqr(dy));
    end
end

```

```

    else
        result:=0;
    End;

END.
UNIT Unit6;
//*****
                INTERFACE
//*****
uses
    Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
    Dialogs, StdCtrls;

type
    TForm6 = class(TForm)
        Edit1: TEdit;
        Label1: TLabel;
        Edit2: TEdit;
        Label2: TLabel;
        Edit3: TEdit;
        Label3: TLabel;
        Label6: TLabel;
        Edit6: TEdit;
        Button1: TButton;
        procedure Button1Click(Sender: TObject);
        procedure FormCreate(Sender: TObject);
    private
        { Private declarations }
    public
        { Public declarations }
    end;

var
    Form6: TForm6;
//*****
                IMPLEMENTATION
//*****
Uses Unit1;
{$R *.dfm}

Procedure TForm6.Button1Click(Sender: TObject);
Begin
    Lmax1:=strtoint(Edit1.Text);
    kp1:=strtoint(Edit2.Text);

```

```

kr1:=strtoint(Edit3.Text);
Mout1:=strtoint(Edit6.Text);
close;
End;

```

```

unit Unit7;
  Form7: TForm7;
//*****
                IMPLEMENTATION
//*****
Uses Unit1;
{$R *.dfm}

```

```

Procedure TForm7.Button1Click(Sender: TObject);
var
  i,j:integer;
Begin
  for i:=0 to StringGrid6.ColCount-1 do
    for j:=0 to 1 do
      yr1[i,j]:=StrToInt(StringGrid6.Cells[i,j+1]);
  for i:=0 to StringGrid7.RowCount-1 do
    q1[i]:=strtofloat(StringGrid7.Cells[1,i]);
  EA.Setf0qset(q1);
  EA.Setq(q1);
  EA.Setyr(yr1);
  Close;
End;

```

```

Procedure TForm7.Button2Click(Sender: TObject);
var
  i,j:integer;
  s:string;
Begin
  Savedialog1.FileName:='yr_'+inttostr(kChoose)+'.txt';
  if Savedialog1.Execute then
  begin
    Memo1.clear;
    Memo1.Lines.Add('*****');
    for i:=0 to StringGrid6.ColCount-1 do
    begin
      s:="";
      for j:=0 to 1 do
        s:=s+StringGrid6.Cells[i,j+1]+' ';
      Memo1.Lines.Add(s);
    end;
  end;

```

```

end;
Memo1.Lines.Add('*****');
for i:=0 to StringGrid7.RowCount-1 do
  Memo1.Lines.Add(StringGrid7.Cells[1,i]);
Memo1.Lines.SaveToFile(Savedialog1.FileName);
Memo1.Clear;
for i:=0 to StringGrid6.ColCount-1 do
begin
  s:="";
  for j:=0 to 1 do
    s:=s+StringGrid6.Cells[i,j+1]+' ';
  Memo1.Lines.Add(s);
end;
Savedialog1.FileName:='yr_'+inttostr(kChoose)+'00.txt';
Memo1.Lines.SaveToFile(Savedialog1.FileName);
Memo1.Clear;
for i:=0 to StringGrid7.RowCount-1 do
  Memo1.Lines.Add(StringGrid7.Cells[1,i]);
  Savedialog1.FileName:='q_'+inttostr(kChoose)+'00.txt';
  Memo1.Lines.SaveToFile(Savedialog1.FileName);
end;
End;

```

```

Procedure TForm7.Button3Click(Sender: TObject);
var
  i,j,k:integer;
  s,s1:string;
Begin
  if OpenFileDialog1.Execute then
  begin
    memo1.Lines.LoadFromFile(OpenDialog1.FileName);
    for i:=0 to StringGrid6.ColCount-1 do
    begin
      s:=memo1.Lines[i];
      k:=1;
      for j:=0 to 1 do
      begin
        s1:="";
        while s[k] in Cyfr do
        begin
          s1:=s1+s[k];
          k:=k+1;
        end;
        if s[k]=' ' then

```

```

        StringGrid6.Cells[i,j+1]:=s1;
        k:=k+1;
        while (k<length(s))and(s[k]=' ') do k:=k+1;
    end;
end;
end;
End;

```

```

Procedure TForm7.Button4Click(Sender: TObject);
var
    i,j:integer;
    s:string;
Begin
    if Opendialog1.Execute then
        begin
            memo1.Clear;
            memo1.Lines.LoadFromFile(Opendialog1.FileName);
            for i:=0 to kR1-1 do
                stringGrid7.cells[1,i]:=memo1.Lines[i];
            end;
        end;
End;

```

```

Procedure TForm7.FormCreate(Sender: TObject);
var
    i,j:integer;
Begin
    color:=RGB(240,240,100);
    StringGrid6.ColCount:=EA.LmaxMout;
    StringGrid7.RowCount:=kR1;
    for i:=0 to EA.LmaxMout-1 do
        StringGrid6.Cells[i,0]:=inttostr(i);

    StringGrid7.ColWidths[1]:=108;
    for i:=0 to StringGrid6.ColCount-1 do
        begin
            StringGrid6.Cells[i,0]:=InttoStr(i);
            for j:=0 to 1 do
                StringGrid6.Cells[i,j+1]:=inttostr(yr1[i,j]);
            end;
        end;
    for i:=0 to kR1-1 do
        begin
            StringGrid7.Cells[0,i]:=InttoStr(i);
            StringGrid7.Cells[1,i]:=floattostrf(q1[i],ffFixed,8,5);
        end;
    end;
End;

```

```
end;
End;
```

```
Procedure TForm7.StringGrid6Click(Sender: TObject);
Begin
  Label7.Caption:='Psi'+inttostr(StringGrid6.Row)+' '+
    inttostr(StringGrid6.Col)+'';
End;

END.
```

```
unit Unit8;
```

```
//*****
```

IMPLEMENTATION

```
//*****
```

```
Uses Unit1;
{$R *.dfm}
```

```
Procedure TForm8.Button1Click(Sender: TObject);
Begin
  hh1:=strtoint(Edit1.Text);
  pp1:=strtoint(Edit2.Text);
  rr1:=strtoint(Edit3.Text);
  nfu1:=strtoint(Edit4.Text);
  lchr1:=strtoint(Edit5.Text);
  kR1:=strtoint(Edit6.Text);
  c1:=strtoint(Edit7.Text);
  d1:=strtoint(Edit8.Text);
  Epo1:=strtoint(Edit9.Text);
  kel1:=strtoint(Edit10.Text);
  gamma1:=strtofloat(Edit11.Text);
  pmut1:=strtofloat(Edit12.Text);
  close;
End;
```

```
Procedure TForm8.FormCreate(Sender: TObject);
Begin
  color:=RGB(250,100,250);
  Edit1.Text:=inttostr(hh1); //number of chromosomes in an initial population
  Edit2.Text:=inttostr(pp1); // number of generations
  Edit3.Text:=inttostr(rr1); // number of couples in one generation
  Edit4.Text:=inttostr(nfu1); // number of functionals
  Edit5.Text:=inttostr(lchr1); // number of variations in one chromosome
```

```

Edit6.Text:=inttostr(kR1); // number of serching parameters
Edit7.Text:=inttostr(c1); // number of bits for integer part
Edit8.Text:=inttostr(d1); // number of bits for fractional part
Edit9.Text:=inttostr(Epo1);//number of generatons between exchange of basic
NOM
Edit10.Text:=inttostr(kel1); // number of elitaring chromosomes
Edit11.Text:=floattostr(gamma1); // parameter for crossover
Edit12.Text:=floattostr(pmut1); // probability of mutation
End;
END.

```

```
unit Unit14;
```

```
//*****
```

IMPLEMENTATION

```
//*****
```

```
Uses Unit1;
{$R *.dfm}
```

```
Procedure TForm14.Button1Click(Sender: TObject);
```

```
var
```

```
  i:integer;
```

```
Begin
```

```
  for i:=0 to EA.ny-1 do
```

```
  begin
```

```
    qymin1[i]:=strtofloat(StringGrid1.Cells[1,i+1]);
```

```
    qymax1[i]:=strtofloat(StringGrid1.Cells[2,i+1]);
```

```
    stepsqy1[i]:=strtofloat(StringGrid1.Cells[3,i+1]);
```

```
  end;
```

```
  Close;
```

```
End;
```

```
Procedure TForm14.FormCreate(Sender: TObject);
```

```
var
```

```
  i:integer;
```

```
Begin
```

```
  Color:=RGB(200,100,250);
```

```
  StringGrid1.RowCount:=EA.ny+1;
```

```
  StringGrid1.ColCount:=4;
```

```
  for i:=1 to EA.ny do
```

```
    StringGrid1.Cells[0,i]:='y['+inttostr(i-1)+']';
```

```
  StringGrid1.Cells[1,0]:='ymin+';
```

```
  StringGrid1.Cells[2,0]:='ymax+';
```

```
  StringGrid1.Cells[3,0]:='delty';
```

```

for i:=0 to EA.ny-1 do
begin
  StringGrid1.Cells[1,i+1]:=floattostrf(qymin1[i],ffFixed,8,4);
  StringGrid1.Cells[2,i+1]:=floattostrf(qymax1[i],ffFixed,8,4);
  StringGrid1.Cells[3,i+1]:=floattostrf(stepsqy1[i],ffFixed,8,4);
end;
close;
End;
END.

unit Unit15;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics,
  Controls, Forms, Dialogs, TeEngine, Series, ExtCtrls, TeeProcs, Chart;

type
  TForm15 = class(TForm)
    Chart1: TChart;
    Series1: TLineSeries;
    Series2: TLineSeries;
    Series3: TLineSeries;
    Series4: TLineSeries;
    Series5: TLineSeries;
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form15: TForm15;
implementation
{$R *.dfm}
end.

UNIT Unit16;
//*****
                IMPLEMENTATION
//*****
{$R *.dfm}
Uses Unit1;
Procedure TForm16.Button1Click(Sender: TObject);

```

```

var
  i,j:integer;
Begin
  Series1.Clear;
  Series2.Clear;
  Series3.Clear;
  Series4.Clear;
  Series5.Clear;
  Series6.Clear;
  Series1.XValues.Order:=loNone;
  Series2.XValues.Order:=loNone;
  Series3.XValues.Order:=loNone;
  Series4.XValues.Order:=loNone;
  Series5.XValues.Order:=loNone;
  Series6.XValues.Order:=loNone;
  for i:=0 to high(xmm[0,0]) do
    Series1.AddXY(xmm[0,0,i],xmm[0,1,i]);
  for i:=0 to high(xmm[1,0]) do
    Series2.AddXY(xmm[1,0,i],xmm[1,1,i]);
  for i:=0 to high(xmm[2,0]) do
    Series3.AddXY(xmm[2,0,i],xmm[2,1,i]);
  for i:=0 to high(xmm[3,0]) do
    Series4.AddXY(xmm[3,0,i],xmm[3,1,i]);
  for i:=0 to 4 do
    Series5.AddXY(prepc[0,i,0],prepc[0,i,1]);
  for i:=0 to 4 do
    Series6.AddXY(prepc[1,i,0],prepc[1,i,1]);
End;

Procedure TForm16.Button2Click(Sender: TObject);
Begin
  if savedialog1.execute then
    Chart1.SaveToBitmapFile(savedialog1.FileName);
End;

Procedure TForm16.FormCreate(Sender: TObject);
var
  i:integer;
Begin
  Color:=RGB(200,170,220);
End;
END.

UNIT Unit17;

```

```
//*****
```

IMPLEMENTATION

```
//*****
```

```
{$R *.dfm}
```

```
Procedure TForm17.Button1Click(Sender: TObject);
```

```
var
```

```
  i,j:integer;
```

```
Begin
```

```
  for i := 0 to nGraphc-1 do
```

```
    for j := 0 to 1 do
```

```
      qyGraph[i,j]:=strtofloat(stringGrid1.Cells[j,i+1]);
```

```
    close;
```

```
End;
```

```
Procedure TForm17.FormCreate(Sender: TObject);
```

```
var
```

```
  i:integer;
```

```
Begin
```

```
  Color:=rgb(200,100,170);
```

```
  StringGrid1.RowCount:=nGraphc+1;
```

```
  StringGrid1.ColCount:=2;
```

```
  for i:=0 to 1 do
```

```
  begin
```

```
    StringGrid1.Cells[i,0]='qy'+inttostr(i+1);
```

```
    StringGrid1.Cells[i,1]:=floattostrf(qymin1[i],ffFixed,6,2);
```

```
    StringGrid1.Cells[i,2]:=floattostrf(qymax1[i],ffFixed,6,2);
```

```
  end;
```

```
  StringGrid1.Cells[0,3]:=floattostrf(qymin1[0],ffFixed,6,2);
```

```
  StringGrid1.Cells[1,3]:=floattostrf(qymax1[1],ffFixed,6,2);
```

```
  StringGrid1.Cells[0,4]:=floattostrf(qymax1[0],ffFixed,6,2);
```

```
  StringGrid1.Cells[1,4]:=floattostrf(qymin1[1],ffFixed,6,2);
```

```
End;
```

```
END.
```

```
UNIT Unit18;
```

```
//*****
```

IMPLEMENTATION

```
//*****
```

```
{$R *.dfm}
```

```
Procedure SetDCPixelFormat (hdc : HDC);
```

```
var
```

```
  pfd : TPixelFormatDescriptor;
```

```

nPixelFormat : Integer;
Begin
  FillChar (pfd, SizeOf (pfd), 0);
  pfd.dwFlags := PFD_DRAW_TO_WINDOW or PFD_SUPPORT_OPENGL or
PFD_DOUBLEBUFFER;
  nPixelFormat := ChoosePixelFormat (hdc, @pfd);
  SetPixelFormat (hdc, nPixelFormat, @pfd);
End;

```

```

Procedure TForm18.FormCreate(Sender: TObject);
Begin
  SetDCPixelFormat(Canvas.Handle);
  hrc := wglCreateContext(Canvas.Handle);
  iTime:=0;
  Hdl:=Ldlc*Clientheight/Clientwidth;
  Ldl:=Ldlc;
  y00:=Hdl/2;
  x00:=Ldl/2;

End;

```

```

Procedure TForm18.FormDestroy(Sender: TObject);
Begin
  wglDeleteContext(hrc);
End;

```

```

Procedure TForm18.FormPaint(Sender: TObject);
Begin
  wglMakeCurrent(Canvas.Handle, hrc);
  glViewport (0, 0, ClientWidth, ClientHeight);
  glClearColor (0.75, 0.75, 0.75, 1.0);
  glClear (GL_COLOR_BUFFER_BIT);
  glColor3f (0.0, 0.0, 0.8);
  glBegin (GL_POLYGON);
  glVertex2f (ym[3,iTime]/Ldl,ym[4,iTime]/Hdl);
  glVertex2f (ym[5,iTime]/Ldl,ym[6,iTime]/Hdl);
  glVertex2f (ym[7,iTime]/Ldl,ym[8,iTime]/Hdl);
  glVertex2f (ym[9,iTime]/Ldl,ym[10,iTime]/Hdl);
  glVertex2f (ym[3,iTime]/Ldl,ym[4,iTime]/Hdl);
  glEnd;
  glColor3f (0.8, 0.0, 0.0);
  glBegin (GL_POLYGON);
  glVertex2f (Prepc[0,0,0]/Ldl,Prepc[0,0,1]/Hdl);
  glVertex2f (Prepc[0,1,0]/Ldl,Prepc[0,1,1]/Hdl);

```

```

glVertex2f (Prepc[0,2,0]/Ldl,Prepc[0,2,1]/Hdl);
glVertex2f (Prepc[0,3,0]/Ldl,Prepc[0,3,1]/Hdl);
glVertex2f (Prepc[0,4,0]/Ldl,Prepc[0,4,1]/Hdl);
glEnd;
glBegin (GL_POLYGON);
glVertex2f (Prepc[1,0,0]/Ldl,Prepc[1,0,1]/Hdl);
glVertex2f (Prepc[1,1,0]/Ldl,Prepc[1,1,1]/Hdl);
glVertex2f (Prepc[1,2,0]/Ldl,Prepc[1,2,1]/Hdl);
glVertex2f (Prepc[1,3,0]/Ldl,Prepc[1,3,1]/Hdl);
glVertex2f (Prepc[1,4,0]/Ldl,Prepc[1,4,1]/Hdl);
glEnd;

```

```

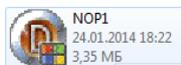
SwapBuffers(Canvas.Handle);
wglMakeCurrent(0, 0);
End;

```

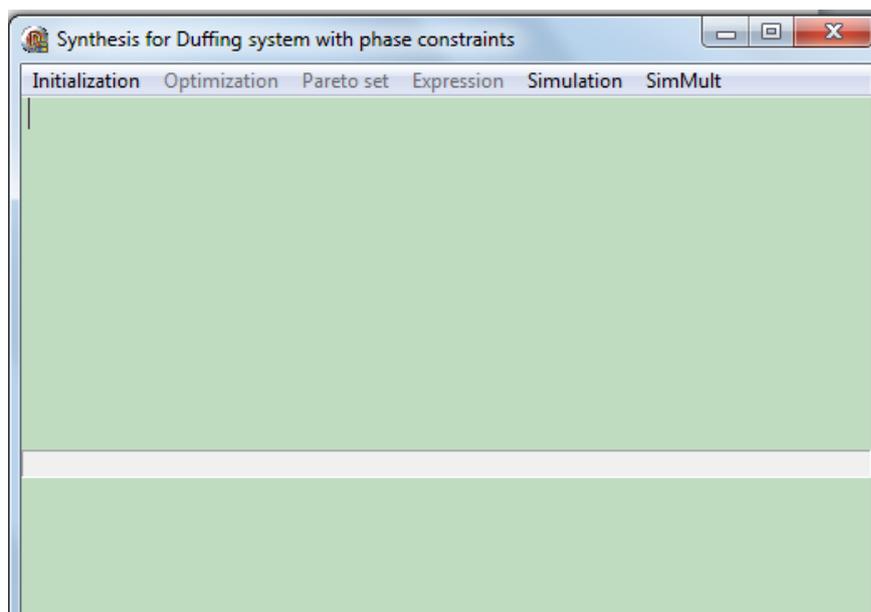
```

Procedure TForm18.Timer1Timer(Sender: TObject);
Begin
  ITime:=(ITime+1) mod (high(tm)+1);
  Refresh;
End;
END.

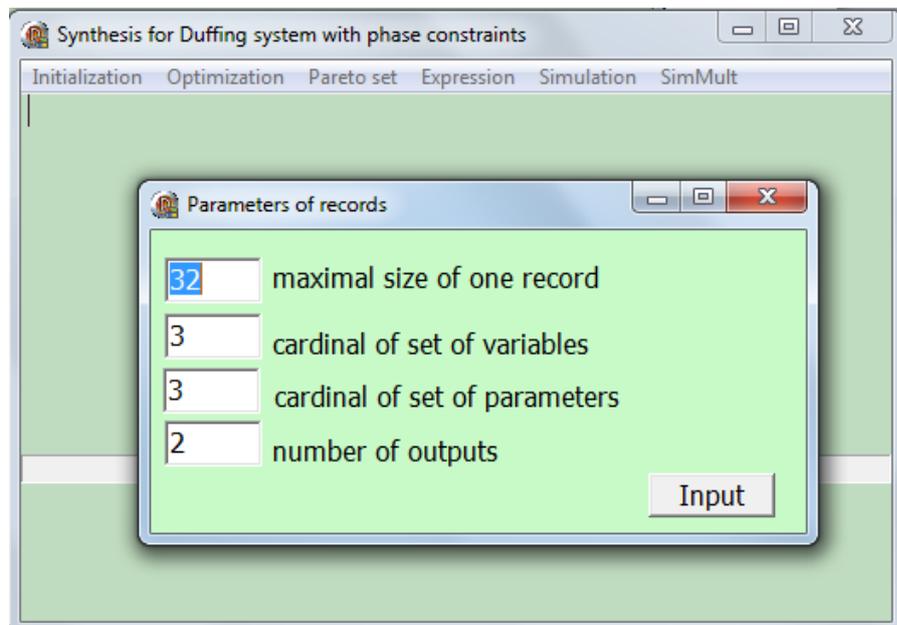
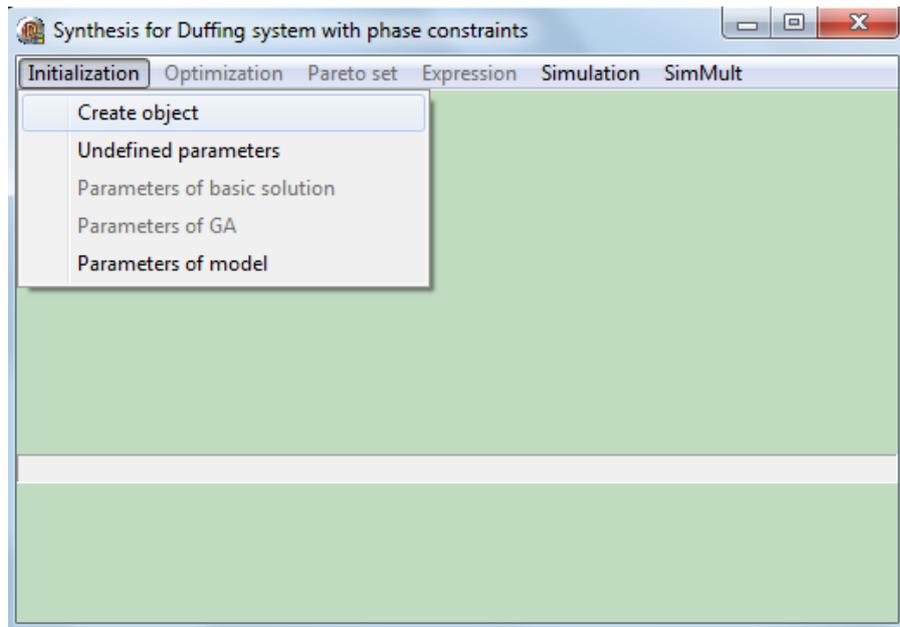
```



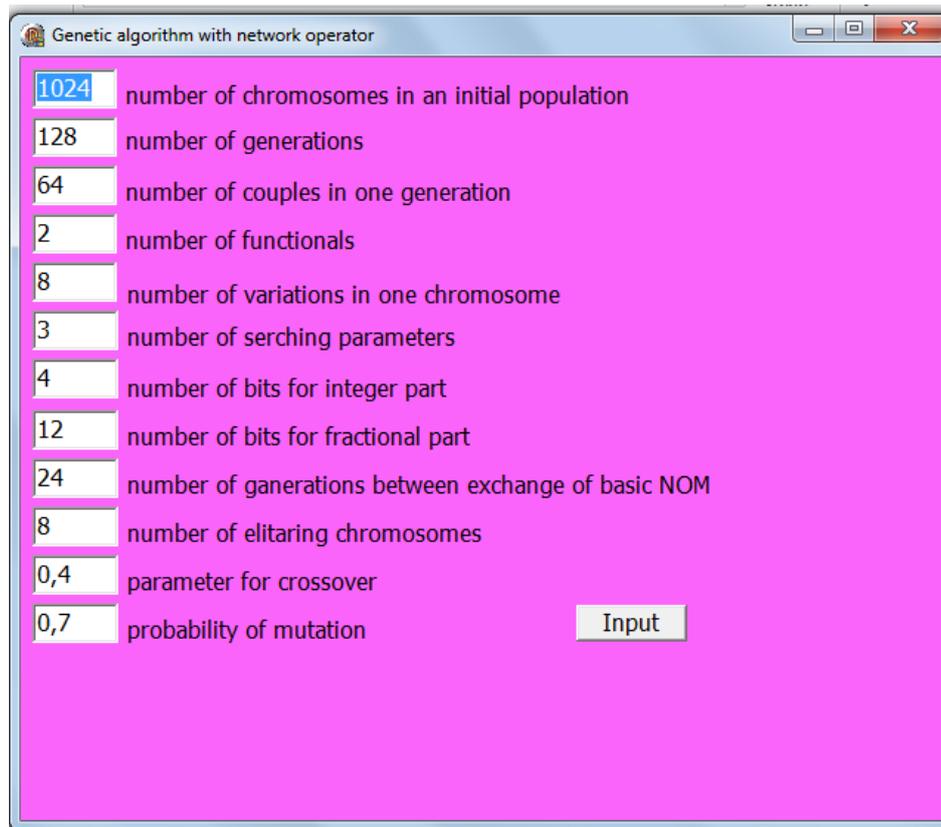
Рабочее окно



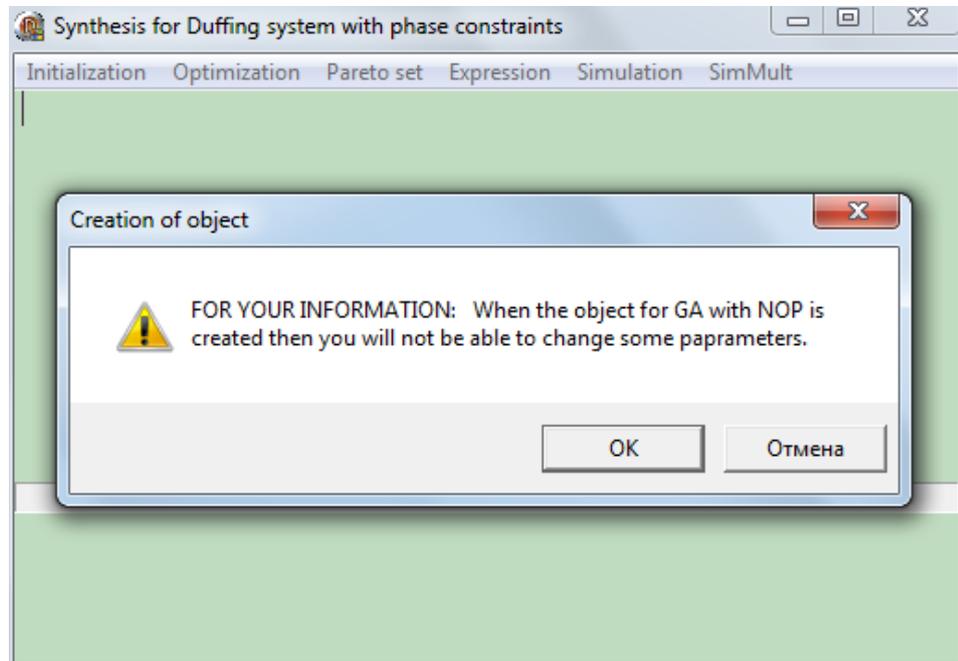
Initialization → Create object(создать объект)



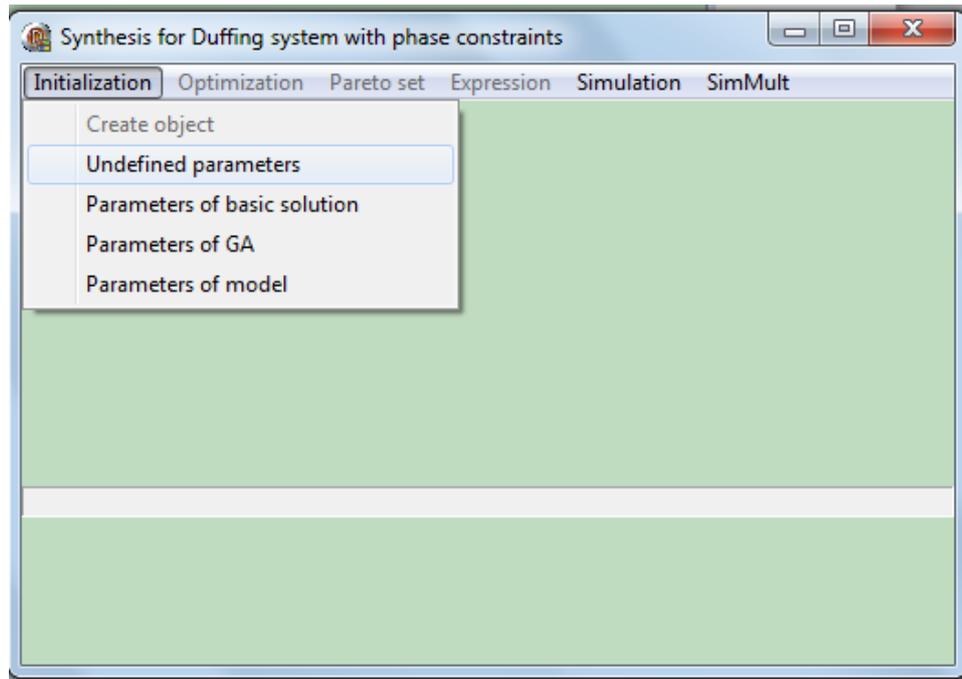
Нажимаем на кнопку Input шаги создания объекта



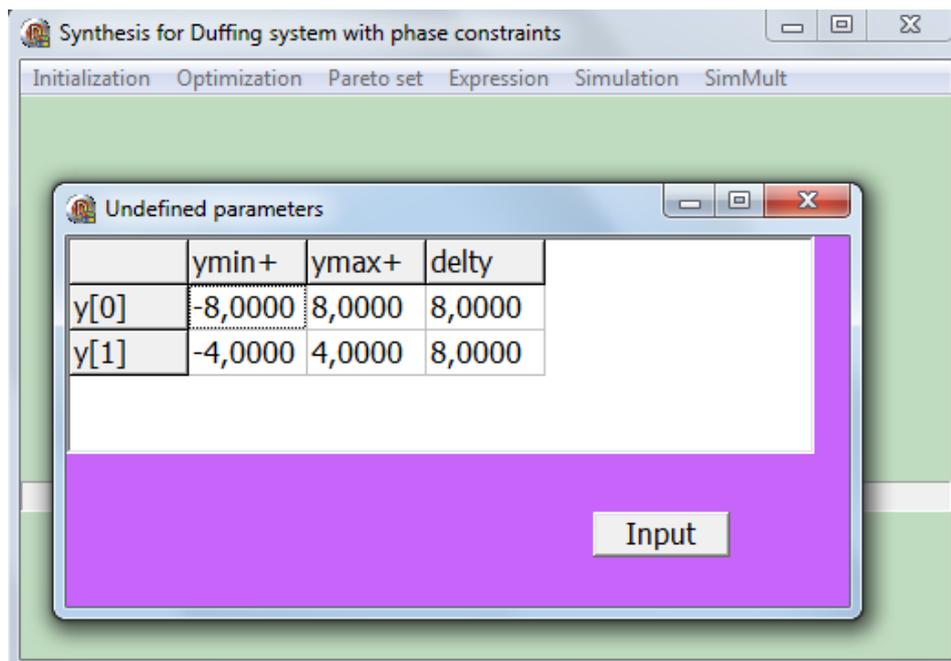
Нажимаем Input



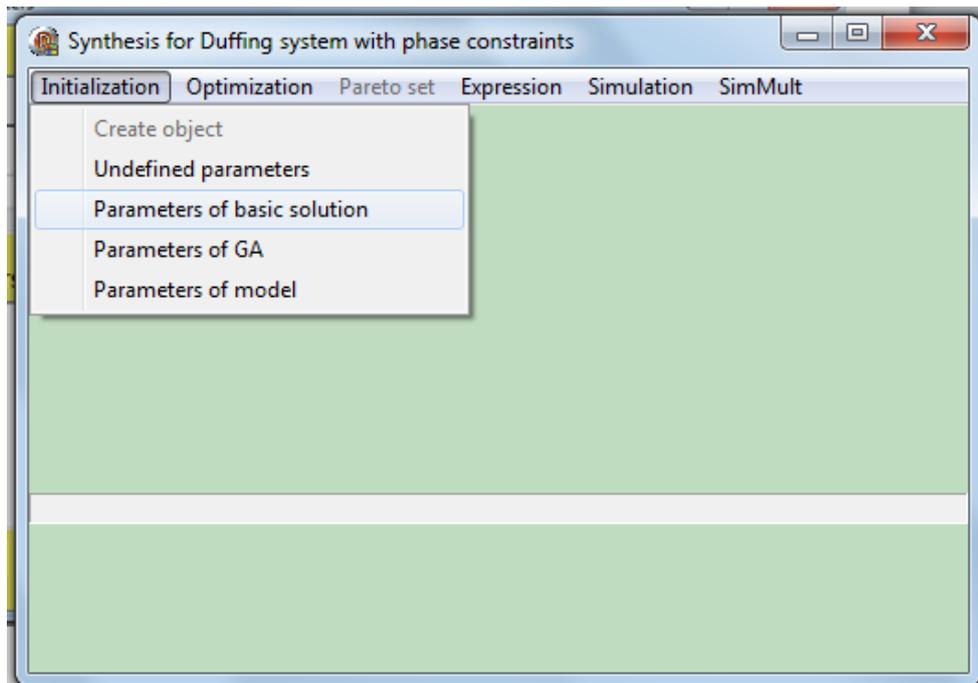
Предупреждение о том, что после создания объекта не как нельзя будет поменять какие то параметры, нажимаем ОК



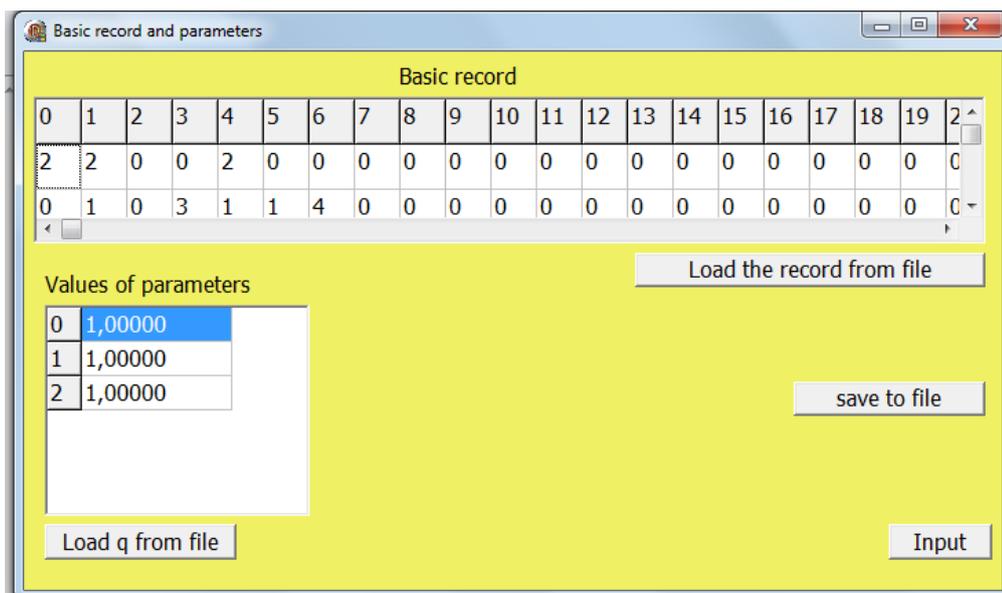
Не определенные параметры



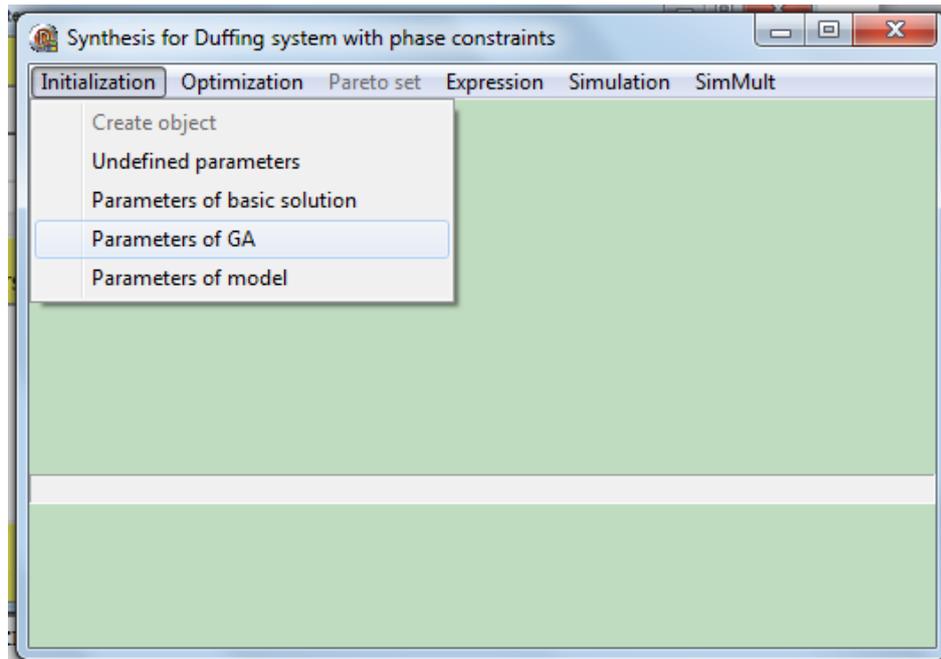
4 начальных условий 2 по x 2 по y, далее Input



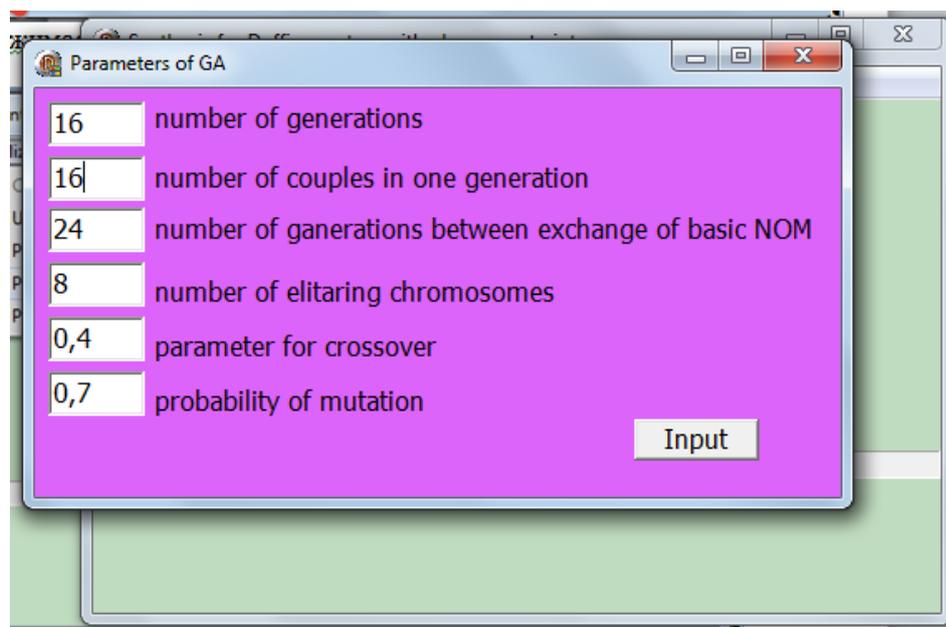
Базисный оператор



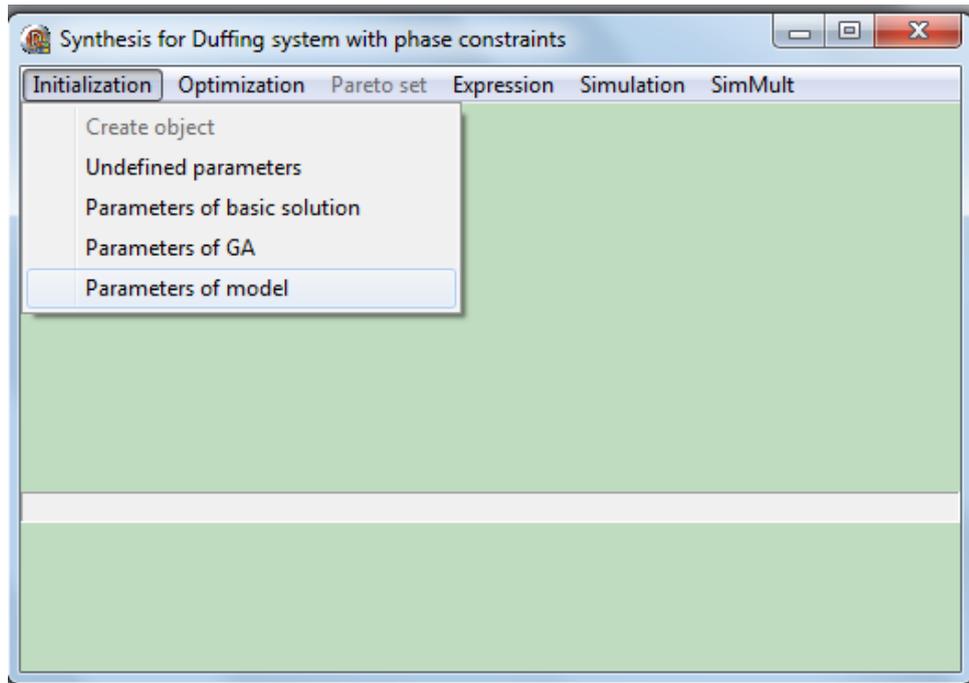
Нажимая на кнопку input загружаем просчитанные данные



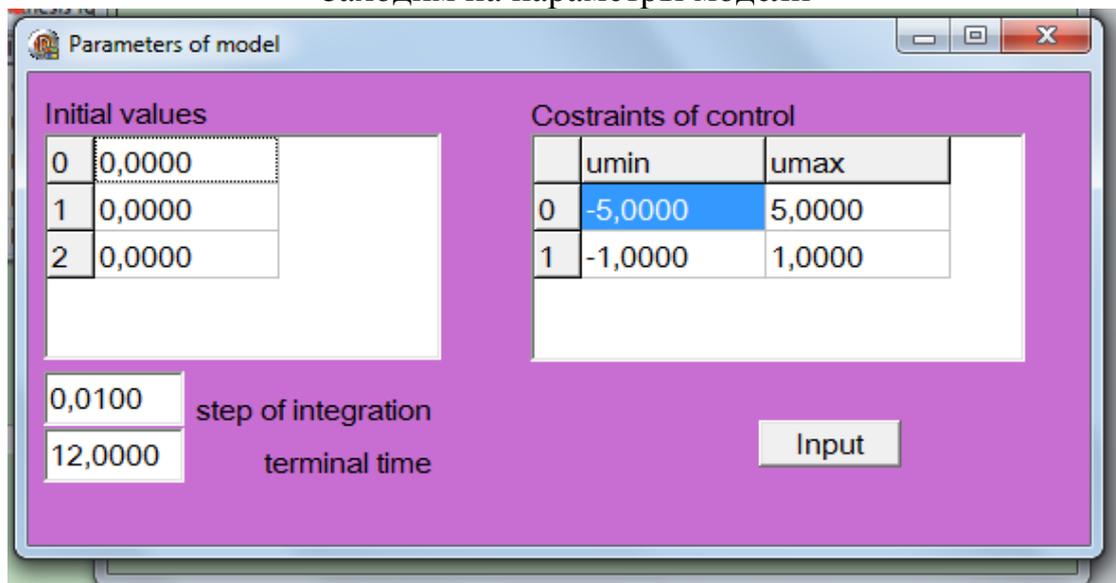
Теперь задаем параметры ГА которые можно менять



Задаем нужные параметры и нажимаем Input

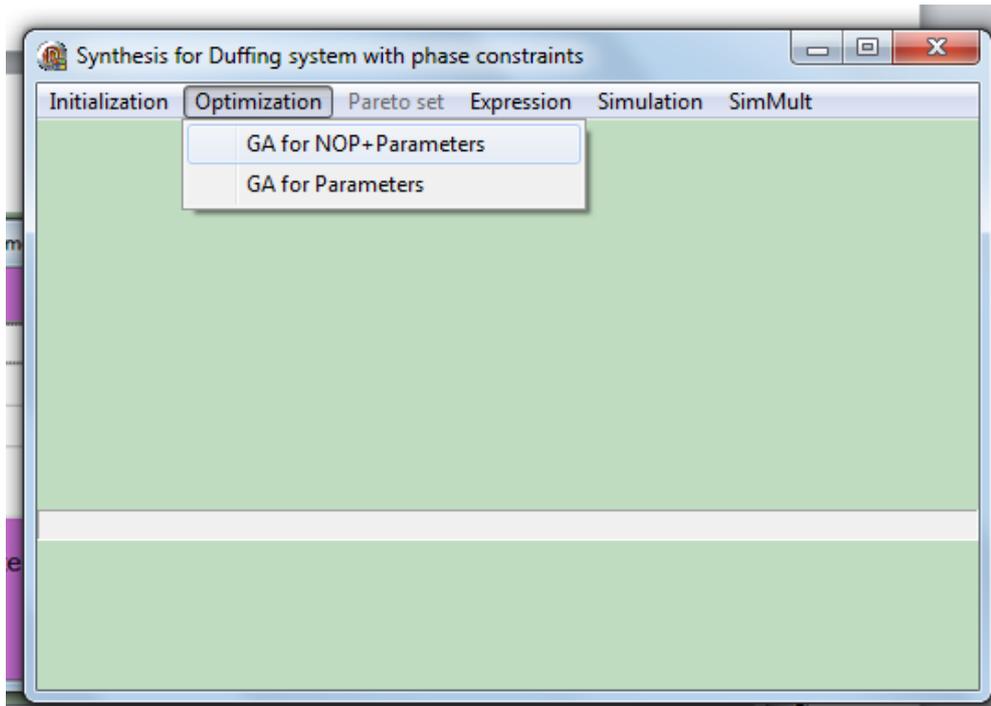


Заходим на параметры модели

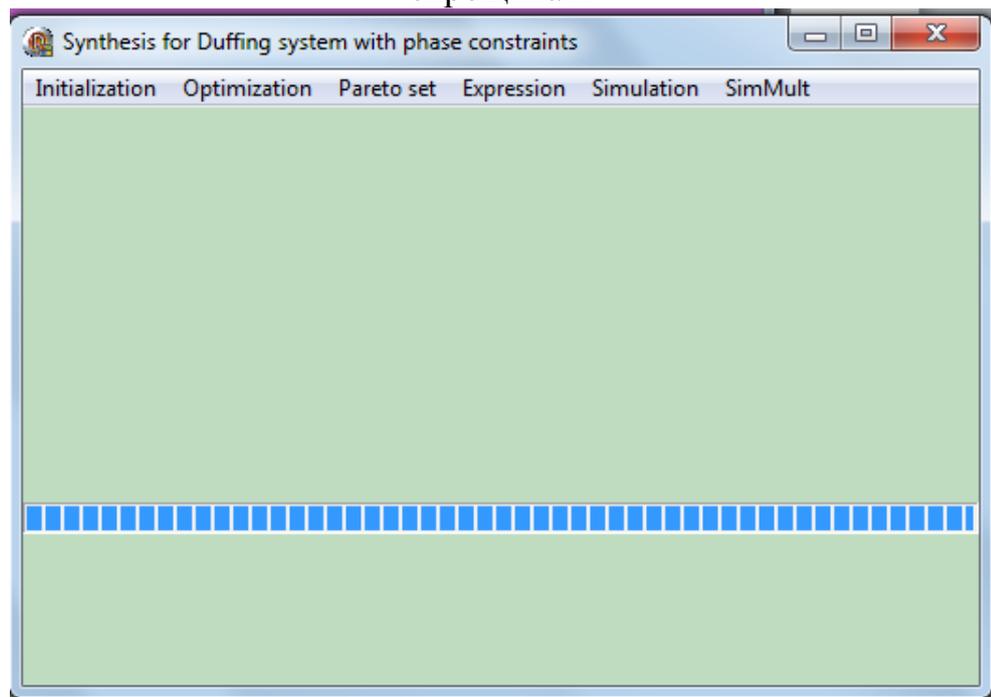


Здесь задан шаг интегрирования, время интегрирования, ограничения по управлению и начальные условия которые могут и добавляться с параметрами.

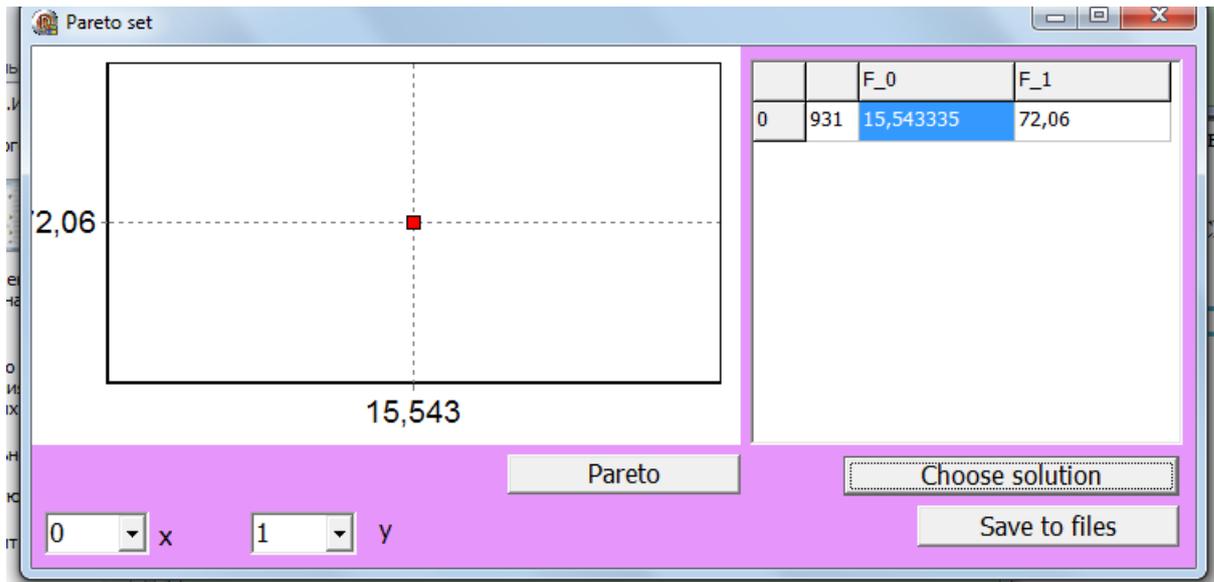
Далее Input



Нажимаем на оптимизацию и он создаст популяцию и по 16 поколениям будет скрещивать



Вот и просчитал, теперь нажимаем на Pareto set входим



Множество Парето, выбираем параметру и нажимаем Choose solution

The 'Initial data' window contains a grid of 20 columns and 3 rows of data. Below the grid, there are two sections for parameter input:

- parameters**

0	5,72437
1	2,35449
2	10,66064
- adaptive parameters**

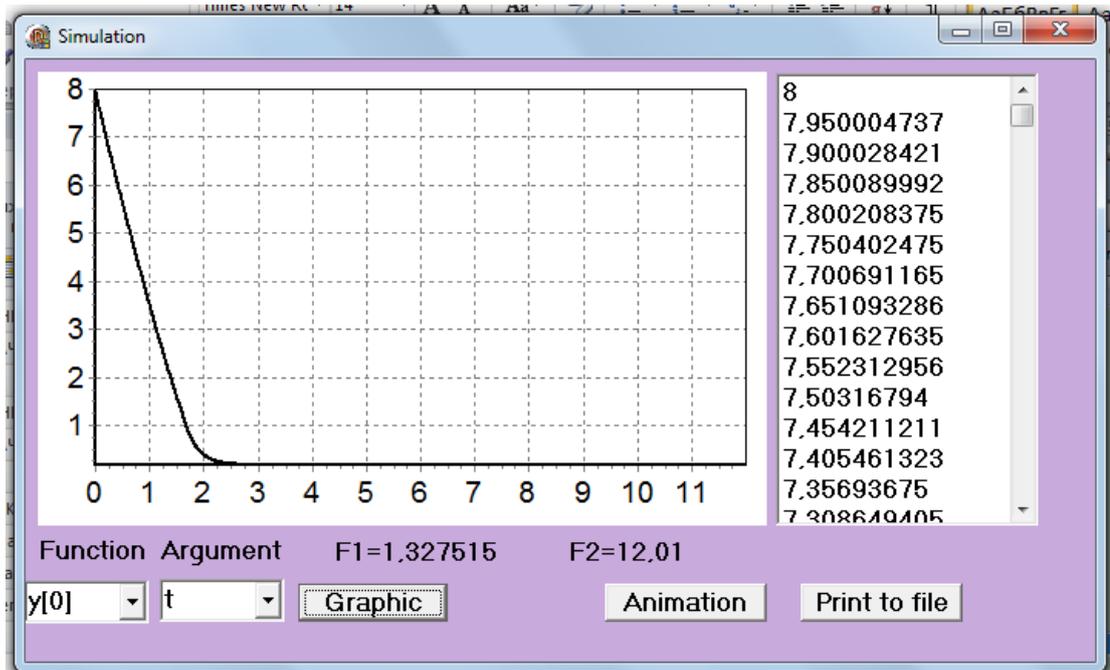
0	8,00000
1	4,00000

Below these sections are three input fields with labels:

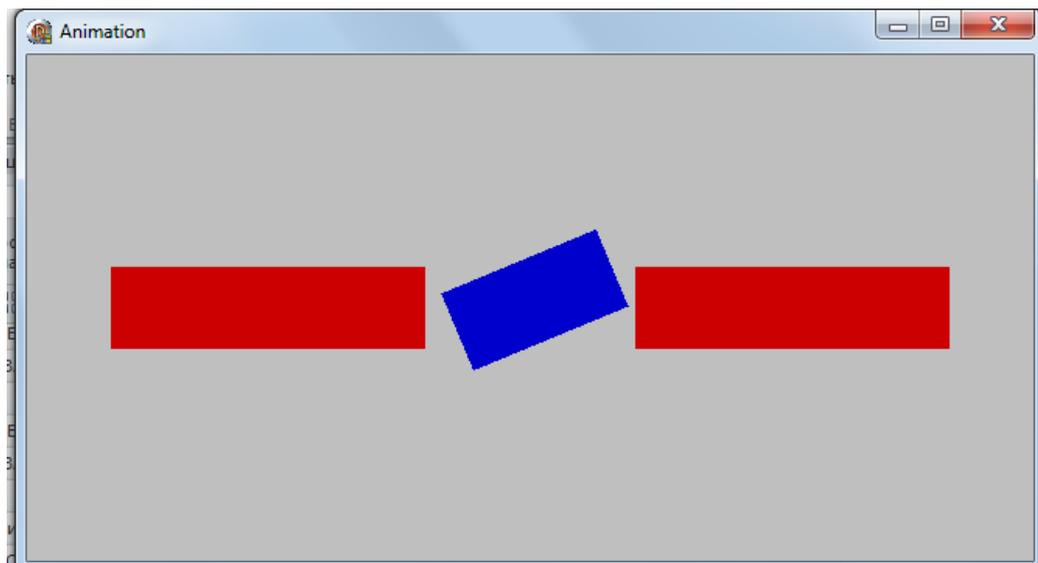
- 0,01 step of integration
- 0,01 step of print
- 12 terminal time

An 'input' button is located at the bottom right of the parameter input area.

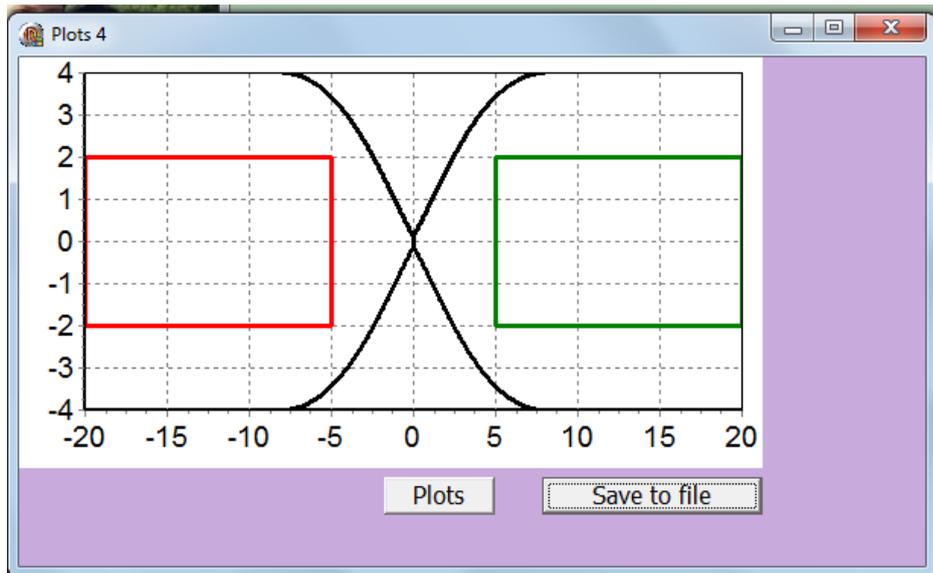
Далее нажимаем input для внедрения параметров



Нажимаем Animation



Вот анимационный вид, паркуется между двумя препятствиями



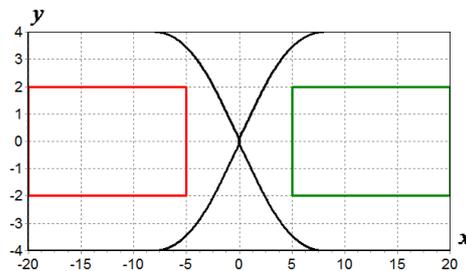
В результате синтеза было получено следующее решение [11,13]

$$S_1 = \left(\begin{bmatrix} 2 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 2 \\ 6 \end{bmatrix}, \begin{bmatrix} 1 \\ 21 \end{bmatrix}, \begin{bmatrix} 1 \\ 7 \end{bmatrix}, \begin{bmatrix} 1 \\ 17 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \begin{bmatrix} 0 \\ 3 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \end{bmatrix} \right),$$

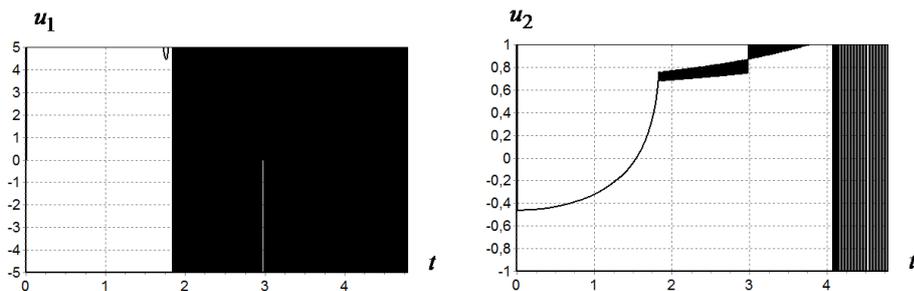
$$S_2 = \left(\begin{bmatrix} 3 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 7 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 10 \end{bmatrix}, \begin{bmatrix} 1 \\ 18 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 14 \end{bmatrix}, \begin{bmatrix} 0 \\ 5 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 17 \end{bmatrix}, \begin{bmatrix} 1 \\ 12 \end{bmatrix}, \right.$$

$$\left. \begin{bmatrix} 1 \\ 19 \end{bmatrix}, \begin{bmatrix} 1 \\ 21 \end{bmatrix}, \begin{bmatrix} 1 \\ 5 \end{bmatrix}, \begin{bmatrix} 1 \\ 20 \end{bmatrix}, \begin{bmatrix} 1 \\ 7 \end{bmatrix}, \begin{bmatrix} 1 \\ 15 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 7 \end{bmatrix}, \begin{bmatrix} 1 \\ 15 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right).$$

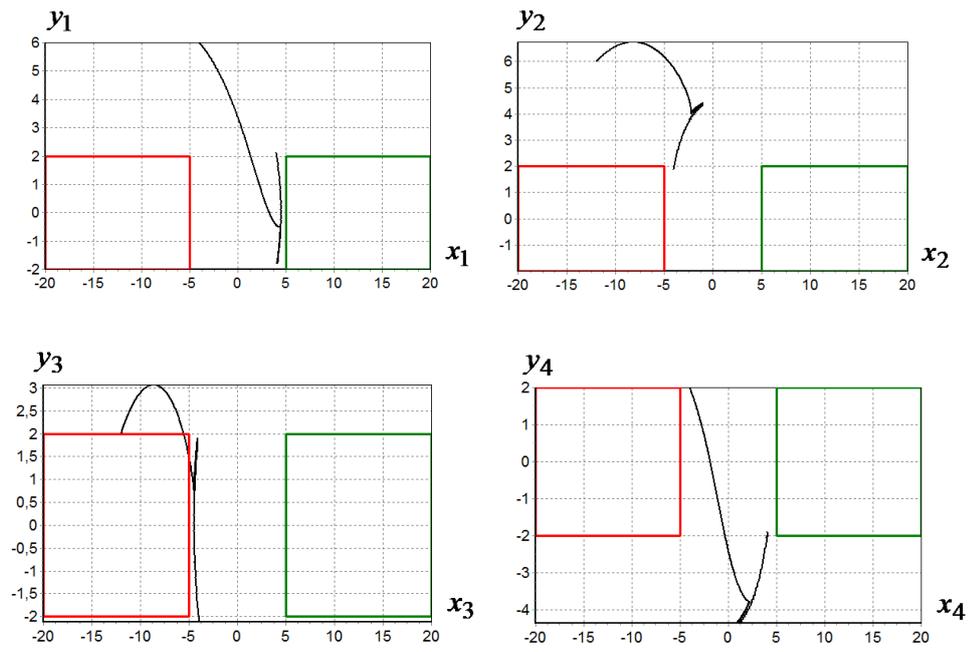
$$q_1 = 12.58398, \quad q_2 = 3.95142, \quad q_3 = 7.75732.$$



Траектории движения центра масс объекта для различных начальных состояний



Графики управлений для начального состояния $x(0) = -8$, $y(0) = 4$, $\theta(0) = 0$



Траектории движения углов объекта для начального состояния

$$x(0) = -8, \quad y(0) = 4, \quad \theta(0) = 0$$