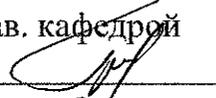


Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
**АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**  
(ФГБОУ ВО «АмГУ»)

Факультет математики и информатики  
Кафедра информационных и управляющих систем  
Направление подготовки 09.04.04 – Программная инженерия  
Направленность (профиль) образовательной программы «Компьютерное моделирование»

ДОПУСТИТЬ К ЗАЩИТЕ

Зав. кафедрой

  
А.В. Бушманов  
« 15 » 07 2020 г.

**МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ**

на тему: Разработка прикладного программного обеспечения для систем адаптивного управления одноканальным объектом с запаздыванием на множестве состояний функционирования

Исполнитель  
студент группы 857 ом



01.07.2020

(подпись, дата)

Д.С. Непомнящий

Руководитель  
профессор, доктор техн. наук



14.07.2020

(подпись, дата)

Е.Л. Ерёмин

Руководитель магистерской программы  
профессор, доктор техн. наук



14.07.2020

(подпись, дата)

И.Е. Ерёмин

Нормоконтроль  
доцент, канд. техн. наук



03.07.2020

(подпись, дата)

В.В. Еремина

Рецензент  
Доцент, канд. физ.-мат. наук



14.07.2020

(подпись, дата)

Д.В. Фомин

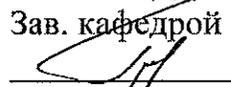
Благовещенск 2020

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
**АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**  
(ФГБОУ ВО «АмГУ»)

Факультет математики и информатики  
Кафедра информационных и управляющих систем

УТВЕРЖДАЮ

Зав. кафедрой

 А.В. Бушманов  
« 02 » 09 2020 г.

### ЗАДАНИЕ

К магистерской диссертации студента Непомнящего Дмитрия Сергеевича.

1. Тема магистерской диссертации: Разработка прикладного программного обеспечения для систем адаптивного управления одноканальным объектом с запаздыванием на множестве состояний функционирования

(утверждено приказом от 30.04.2020 № 810 - уч)

2. Срок сдачи студентом законченной работы 20.06.2020

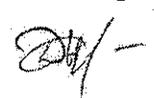
3. Исходные данные магистерской диссертации: отчет о прохождении преддипломной практики.

4. Содержание магистерской диссертации: описание предметной области, синтез адаптивной системы управления, имитационное моделирование, разработка прикладного программного обеспечения.

5. Перечень материалов приложения: построение эквивалентной системы, Simulink-схема системы при имитационном моделировании, Основные этапы разработки спиральной модели, функциональная диаграмма прикладного программного обеспечения, UML-диаграммы разрабатываемого программного обеспечения, блок-схема алгоритма расчета имитационных характеристик.

6. Дата выдачи задания: 02.09.2019

Руководитель магистерской диссертации: Ерёмин Евгений Леонидович, профессор, доктор техн. наук.

Задание принял к исполнению 02.09.2020  Д.С. Непомнящий

## РЕФЕРАТ

Магистерская диссертация содержит 80 с., 27 рисунков, 6 приложений, 24 источника.

АДАПТИВНАЯ СИСТЕМА УПРАВЛЕНИЯ, АДАПТИВНЫЙ РЕГУЛЯТОР, АПРИОРНАЯ НЕОПРЕДЕЛЕННОСТЬ, ВИДОИЗМЕНЕННЫЙ ОБЪЕКТ УПРАВЛЕНИЯ, ДИНАМИЧЕСКИЙ КОРРЕКТОР, ГИПЕРУСТОЙЧИВОСТЬ, L-ДИССИПАТИВНОСТЬ, ЗАПАЗДЫВАНИЕ ПО СОСТОЯНИЮ, ЯВНО-НЕЯВНАЯ ЭТАЛОННАЯ МОДЕЛЬ, МНОЖЕСТВО СОСТОЯНИЙ ФУНКЦИОНИРОВАНИЯ.

Цель работы – разработка прикладного программного обеспечения для анализа работоспособности синтезированной адаптивной системы управления классом объектов с несколькими состояниями функционирования.

Объект исследования – теория адаптивного управления объектом на множестве состояний функционирования.

Результатом работы является прикладное программное обеспечение для работы с синтезированной адаптивной системой управления одноканальным объектом с запаздыванием на множестве состояний функционирования.

## СОДЕРЖАНИЕ

Введение	8
1 Описание предметной области	11
1.1 Понятие управления	11
1.2 Принципы управления	12
1.2.1 Принцип программного управления	12
1.2.2 Принцип компенсации	13
1.2.3 Принцип обратной связи	13
1.2.4 Принцип комбинированного управления	14
1.3 Виды объектов управления	14
1.4 Управляемость	16
1.5 Наблюдаемость	17
1.6 Методы управления объектами в условиях неопределенности	20
1.7 Методы исследования устойчивости адаптивных систем	25
1.8 Этапы синтеза систем по критерию гиперустойчивости Попова	30
2 Синтез адаптивной системы управления	32
2.1 Математическое описание объекта управления	32
2.2 Постановка задачи	35
2.3 Синтез алгоритмов системы адаптации	36
3 Имитационное моделирование	40
3.1 Описание среды моделирования	40
3.1.1 Обзор сред моделирования	40
3.1.2 Среда моделирования Matlab	42
3.1.3 Описание модуля Simulink	43
3.2 Задание параметров системы	45
3.3 Характеристики системы	46
4 Разработка прикладного программного обеспечения	52
4.1 Обоснование разработки прикладного программного обеспечения	52

4.2	Выбор модели жизненного цикла программного обеспечения	53
4.3	Описание функций программного обеспечения	58
4.4	Архитектурный проект программного обеспечения	61
4.4.1	Диаграмма вариантов использования	61
4.4.2	Диаграмма последовательности	62
4.4.3	Диаграмма состояний	63
4.4.4	Диаграмма активности	65
4.4.5	Диаграмма классов	66
4.5	Описание реализации прикладного программного обеспечения	67
4.5.1	Описание среды разработки приложения	67
4.5.2	Реализация взаимодействия с ядром Matlab	68
4.5.3	Реализация алгоритма расчета имитационных характеристик системы	70
4.5.4	Реализация взаимодействия приложения с базой данных	71
4.6	Руководство пользователя	72
4.6.1	Авторизация	72
4.6.2	Рабочая область	72
4.6.3	Настройка параметров моделирования	75
4.6.4	Просмотр результатов моделирования	77
	Заключение	81
	Библиографический список	83
	Приложение А Построение эквивалентной системы	86
	Приложение Б Simulink-схема системы при имитационном моделировании	89
	Приложение В Основные этапы разработки спиральной модели	92
	Приложение Г Функциональная диаграмма прикладного программного обеспечения	93
	Приложение Д UML-диаграмма разрабатываемого программного обеспечения	95
	Приложение Е Блок схема алгоритма расчета имитационных характеристик	100

## НОРМАТИВНЫЕ ССЫЛКИ

В настоящей магистерской диссертации использованы следующие нормативные ссылки на государственные стандарты.

ГОСТ 19.001 – 77 Единая система программной документации. Общие положения

ГОСТ 19.001 – 77 ЕСПД Общие положения

ГОСТ 19.102 – 77 Единая система программной документации. Стадии разработки

ГОСТ 19.201 – 78 Единая система программной документации. Техническое задание. Требования к содержанию и оформлению

ГОСТ 2.104 – 2006 ЕСКД Основные надписи

ГОСТ 2.105 – 95 ЕСКД Общие требования к текстовым документам

ГОСТ 2.106 – 96 ЕСКД Текстовые документы

ГОСТ 2.701 – 84 ЕСКД Схемы. Виды и типы. Общие требования к выполнению

ГОСТ 3.1116 – 79 ЕСКД Нормоконтроль

ГОСТ 7.05 – 2008 Система стандартов по библиотечному и издательскому делу. Библиографическая ссылка. Общие требования и правила составления

ГОСТ 7.12 – 93 Система стандартов по информации, библиотечному издательскому делу. Библиографическая запись. Сокращение слов на русском языке

ГОСТ 7.9 – 95 Система стандартов по информации, библиотечному и издательскому делу. Реферат и аннотация. Общие требования

## ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ, СОКРАЩЕНИЯ

В рамках данной магистерской диссертации были использованы следующие определения, обозначения и сокращения:

Simulink-схема – один из существующих видов графического представления моделей, описывающих элементарные звенья объекта и связи между ними;

UML-диаграмма – схема в графическом представлении, служащая для описания объектного моделирования для разработки ПО;

SOM – технология использующая компонентную модель объекта (англ. component object model);

MySQL – свободная реляционная система управления базами данных;

Рабочая станция – комплекс средств как аппаратных, так и программных, необходимых и достаточных для решения определенных задач;

ВОУ – видоизмененный объект управления;

ВДК – видоизмененный динамический корректор;

ДК – динамический корректор;

ИНП – интегральное неравенство Попова;

ЛСЧ – линейная стационарная часть;

ННЧ – нелинейная нестационарная часть;

ОУ – объект управления;

СУБД – система управления базами данных;

ЭМ – эталонная модель;

ЯЭМ – явная эталонная модель;

ЯНЭМ – явно неявная эталонная модель.

НЭМ – неявная эталонная модель.

## ВВЕДЕНИЕ

Разработка прикладного программного обеспечения для систем робастного, адаптивного и других видов управлений является одной из востребованных и стремительно развивающихся видов деятельности. Программное обеспечение необходимо как для эксплуатации систем управления, так и для имитационного моделирования, позволяющего изучить поведение системы в разных условиях эксплуатации перед введением системы в работу. В рамках данной магистерской диссертации будет синтезирована адаптивная система управления и спроектировано программное обеспечение для анализа работоспособности класса адаптивных систем управления объектами с несколькими состояниями функционированиями.

Теория управления является наукоемкой и серьезной дисциплиной, исследующей различные способы и виды управления технологическими объектами. Данная дисциплина получила свое развитие при появлении примитивных электромашин и радиоавтоматики, поскольку потребовались инструменты, позволяющие эффективно использовать новейшие изобретения. С каждым годом теория автоматического управления применяется практически в каждом технологическом процессе, поскольку каждое крупное предприятие не может эффективно функционировать без автоматизации какого-либо процесса не зависимо от природы. Так же каждый автоматизированный процесс требует наличие соответствующего программного обеспечения для выполнения различных функций: тестирования, контроля и управления текущим процессом.

С усложнением процессов и устройств автоматизации традиционные методы теории управления утратили актуальность, поскольку становится невозможным сформировать точную математическую модель ОУ. Ввиду усложнения ОУ методы теории управления так же развивались, появились такие методы управления как адаптивное, робастное, нейросетевых регуляторов и методов использующих нечеткую логику.

Данное развитие позволило синтезировать и строить системы управления с наличием неопределенности различного типа, а именно: сигнальной неопределенности, неопределенности параметрической, а также функциональной.

Основным преимуществом данных систем, является получение необходимых сведений об объекте уже в процессе функционирования системы, что безусловно выводит на новый уровень теорию адаптивного управления. При этом адаптивное управления позволяет решать задачи управления объектами, имеющими неопределенность в математическом описании, а также наличии различного рода запаздывании, возмущении внешней среды, а также наличии у объектов сложной динамики, в частности, наличие нескольких состояний функционирования [2-6]. К таким системам относятся системы с переключениями, используемые в задачах управления технологическими, энергетическими и механическими объектами. Рассматриваемые системы состоят из набора коммутируемых подсистем, имеющих общий закон переключения, который определяет какая подсистема активна в данный момент времени.

Таким образом, адаптивное управления позволяет решать задачи управления сложными объектами, максимально приближенными к реальным условиям эксплуатации, при этом, данные системы обеспечивают желаемое качество работоспособности системы.

Объектом исследования данной магистерской диссертации является система управления объектом, имеющим несколько состояний функционирования.

Цель работы – разработка прикладного программного обеспечения для анализа работоспособности синтезированной адаптивной системы управления классом объектов с несколькими состояниями функционирования.

В рамках магистерской диссертацией были выделены следующие задачи:

- произвести анализ существующих методов синтеза требуемой адаптивной системы управления;
- на основе выбранного метода синтезировать требуемую адаптивную систему управления;

– исследовать качество работы синтезированной адаптивной системы управления путем имитационного моделирования;

– разработать прикладное программное обеспечение для работы с синтезированной адаптивной системой управления.

В результате достижения цели данной магистерской диссертации должно быть разработано прикладное программное обеспечение для анализа и настройки адаптивной системы управления для класса объектов с несколькими состояниями функционирования.

Практическая значимость магистерской диссертации заключается в разработке прикладного ПО, позволяющего работать с адаптивными системами управления объектами, имеющими конечное число состояний функционирования. Кроме того, в ходе выполнения данной работы будут получены и закреплены знания в области синтеза адаптивных систем управления. Получены знания и опыт в разработке прикладного программного обеспечения.

Согласно цели и задачам, определенным для данной магистерской диссертации, была задана следующая структура.

Введение содержит тезисное описание предмета и объекта исследования. Данная часть работы раскрывает актуальность тематики, кроме того, исследуется практическая и теоретическая значимость.

Первая глава работы описывает существующие методы синтеза адаптивных систем управления, производится анализ подходящего решения для текущей задачи.

Во второй части задается точное математическое описание элементов системы управления, формулируется постановка задачи и производится синтез адаптивной системы управления.

Третья глава содержит информацию о работоспособности синтезированной системы управления на основе имитационного моделирования. Анализируется качество работы контура адаптации.

В заключительной главе исследуется необходимость разработки прикладного программного обеспечения для построенной системы.

# 1 ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ

## 1.1 Понятие управления

Процесс управления – процесс, функционирующий как в живой, так и неживой природе, заключающийся в приеме (получении), хранении, преобразовании информации и выработке на основе собранных данных требуемого управления. В технике управлением называют целенаправленное воздействие на какое-либо устройство или объект, с целью получения требуемого результата. Управление подразделяется на несколько видов, ручное управление – неавтоматическое и управление без участия человека – автоматическое. Важно отметить, что любое устройство (технологический процесс или машина), состоянием которого нужно управлять, называется объектом управления. Как правило, целью любого управления служит поддержание заданного режима работы, в котором параметры, характеризующие состояние объекта управления, изменяются по определенному закону, либо устанавливаются в требуемую величину с заданной погрешностью. Одним из простейших примеров объекта управления является резервуар с жидкостью, в котором требуется поддерживать жидкость на заданном уровне. В этом случае управляемой переменной является уровень воды в резервуаре, а регулирующим устройством – вентиль на входной трубе.

Объект управления, взаимодействующий с управляющим устройством, называют системой управления. Такие системы так же подразделяются на автоматизированные системы управления (АСУ) – системы в которых непосредственное участие принимает человек, и системы автоматического управления (САУ) – системы без непосредственного участия человека.

Объект управления в зависимости от реакции на входные воздействия может быть устойчивым, нейтральным или неустойчивым.

Допустим, что при входном воздействии  $u = u_0, f = f_0$ , выход объекта  $y = y_0$ , если на какой-то промежуток времени  $T$  одно из входных воздействий изменится  $u = u_0 + \nabla$ , а затем примет первоначальное значение  $\nabla = 0$ , при этом  $y(t) \rightarrow y_0$  при  $t \rightarrow \infty$ , то такой объект будет называться устойчивым. В случае, если по-

сле кратковременного изменения входного параметра выход объекта управления будет стремиться к постоянному значению, отличному от первоначального  $y(t) \rightarrow y^* \neq y_0$  при  $t \rightarrow \infty$ , то объект управления будет называться нейтральным. Неустойчивым объект управления будет называться, если выходная переменная не стремится к первоначальному или новому постоянному значению.

## 1.2 Принципы управления

Решение большинства задач теории управления строится на базисных принципах. На данный момент существует четыре основополагающих принципа, позволяющих спроектировать систему автоматического управления любой сложности. К таким принципам относятся: принцип программного управления, принцип компенсации, принцип обратной связи, принцип комбинированного управления. Для понимания основной идеи каждого принципа, изучим их подробнее [13].

### 1.2.1 Принцип программного управления

Данный принцип управления применяется в случае, если рассматривается наиболее простая система, когда объект управления имеет точную математическую модель без неопределенностей. Используя известный закон изменения выходной характеристики ОУ от задающего воздействия, становится возможным построить функцию времени, компенсирующую динамику объекта, тем самым, приведя выходную характеристику к требуемому эталону. При таком принципе управления управляющее устройство можно представить в виде программатора и исполнительного устройства. На рисунке 1 представлена блок-схема системы программного управления:

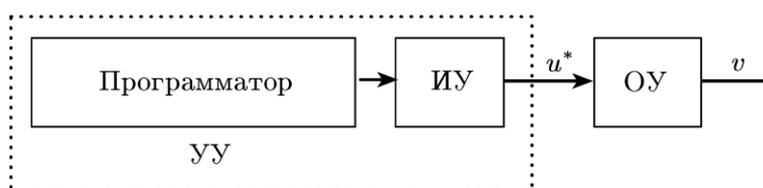


Рисунок 1 – Система программного управления

Принцип программного управления невозможно применять в случае, когда на объект управления действует неизвестное возмущение, оказывающее существенное влияние на управляемую величину. Так же данный метод нельзя

использовать, если объект управления является нейтральным или неустойчивым и систему управления должна функционировать достаточно длительное время. Это связано с тем, что систематическая ошибка в программном управлении приводит к нарастающей ошибке управляемой переменной. В современной теории управления данный метод используется только в очень простых технологических процессах, в основном метод считается устаревшим.

### 1.2.2 Принцип компенсации

При воздействии на объект управления различного рода возмущений, оказывающих существенное влияние на ее работу сложность выработки закона управления для программатора может быть несоизмеримо сложным. В этом случае, при возможности измерения всех возмущений, оказывающих существенное влияние на объект управления, возможно выработать управляющее воздействие, которое полностью или частично компенсирует влияние возмущений на процесс управления. Способ управления, при котором управляющее воздействие вырабатывается на основе действующих возмущений, называется способом управления по возмущению или принципом компенсации.

Главным достоинством данного принципа является возможность полной компенсации возмущающего воздействия.

Недостатком метода является то, что он не применим, если возмущения невозможно измерить. Так же невозможно использовать данный метод, в случае управления нейтральным или неустойчивым объектом.

### 1.2.3 Принцип обратной связи

Управление по отклонению называется такой способ управления, при котором определяется отклонение текущего значения выходной переменной от требуемого значения и на его основе формируется управляющее воздействие.

Системы управления, основанные на способе управления по отклонению, являются замкнутыми: они непременно содержат обратную связь – канал связи, по которому информация об управляемой переменной поступает на управляющее устройство. Поэтому способ управления по отклонению называют также принципом обратной связи.

Достоинством принципа обратной связи является его универсальность, возможность его использования в условиях отсутствия информации о возмущающих воздействиях.

Недостатком способа управления по отклонению является принципиальная невозможность полной компенсации возмущающих воздействий. Это связано с тем, что при этом способе управления управляющее воздействие начинает вырабатываться и оказывать влияние на ход процесса управления только после того, как возмущение, начав действовать, вызывает отклонение управляемой величины от требуемого режима. Кроме того, следует отметить, что система управления обратной связью может оказаться неустойчивой, хотя объект управления устойчив.

#### 1.2.4 Принцип комбинированного управления

Выше при рассмотрении принципа компенсации и принципа обратной связи отмечалось их достоинства и недостатки. Естественной является попытка поиска способа управления, в котором бы совмещались достоинства обоих принципов управления. Таким способом управления является принцип комбинированного управления, при котором одновременно используются способы управления как по возмущению, так и по отклонению.

Данный способ управления применяется в случаях, когда на систему действует много различных возмущений, один (или несколько) из которых оказывает наибольшее влияние на работу системы управления и может быть измерен. В подобных случаях влияние превалирующего возмущения можно нейтрализовать, используя принцип компенсации, а влияние остальных возмущений нейтрализовать, используя принцип обратной связи.

### 1.3 Виды объектов управления

Объект управления – обобщающий термин, описывающий динамический процесс или устройство, управления поведением которого является основной целью при создании систем автоматического управления. На данный момент существуют разнообразные виды объектов управления. Рассмотрим одну из обобщенных классификаций по ряду признаков:

- по виду значения выходной характеристики ОУ;
- по виду функционирования технологического процесса;
- по количеству входных и выходных величин и их взаимосвязи;
- по структуре ОУ;
- по виду стационарности ОУ;
- по виду статических характеристик и характеру математических отношений;
- по реакции ОУ на внешние возмущающие воздействия;

Рассмотрим подробнее каждый тип объекта управления. При классификации объектов управления в соответствии с видом значения выходной характеристики в установившемся режиме, возможно выделить два типа: с самовыравниванием и без самовыравнивания. Самовыравнивание – это процесс, при котором объект может вернуться в начальное состояние при нарушении равновесия между входной и выходной величины.

В зависимости от технологического процесса, ОУ может быть непрерывными либо циклическим.

При классификации объектов управления относительно структуры возможно выделить ОУ имеющие запаздывание и ОУ не имеющие запаздывания. Запаздывание – свойство характеризующиеся различной скоростью распространения информации в среде функционирования ОУ. Данное свойство свойственно большинству действующих технологических процессов.

Классификация в зависимости от количества входных величин и их взаимосвязей подразделяет объекты на: одномерные, а также многомерные многосвязные и многомерные несвязные объекты. Одномерным ОУ называется объект, имеющий только одну выходную величину, многомерные объекты имеют сложную структуру подсистем, позволяющих получать более одного выходного сигнала. Свойство многосвязности описывает взаимосвязь входных и выходных величин, в случае, когда объект многосвязан, то изменение одной входной величины повлечет изменение нескольких выходных.

При классификации объектов по виду статических характеристик все ОУ подразделяются на линейные ОУ и нелинейные ОУ. Под линейностью ОУ понимается возможность описать поведение линейными функциями. Нелинейные объекты управления напротив содержат нелинейности в математическом описании.

При рассмотрении объектов относительно свойства стационарности все объекты подразделяются на нестационарные и стационарные. Свойство стационарности предполагает неизменность параметров объекта в процессе функционирования. Одним из известных примеров нестационарного ОУ является летательный аппарат, использующий топливо. В процессе движения топливо выгорает, что способствует снижению массы аппарата.

При рассмотрении объектов в зависимости от интенсивности случайных возмущений, которые действуют на объект выделяют объекты стохастические и детерминированные. Стохастические объекты управления являются наиболее сложными, поскольку имеют в математической модели случайные факторы, воздействующие на динамику. Существуют различные методы исследования таких объектов, но универсального принципа управления не существует.

#### **1.4 Управляемость**

Основной целью теории управления является получение такого управляющего сигнала, который приведет любой рассматриваемый объект к требуемой процессом динамики. Данная цель полностью потеряет смысл, в случаях, когда ОУ не поддается управления. Для проверки возможности формирования управляющего сигнала для ОУ было сформулировано важное свойство теории управления - управляемость.

Для ввода в понятия в понятие управляемости, рассмотрим один из простых примеров математического описания ОУ:

$$\frac{dx}{dt} = f(x, u, t), \quad x = \text{col}(x_1, \dots, x_n), \quad u = \text{col}(u_1, \dots, u_m), \quad (1)$$

рассматриваемый объект будет являться управляемым, если для каждой из точек  $x(t_{нач})$  и  $x(t_{кон})$  существует такое управляющее воздействие  $u(t)$  на всем вре-

менном интервале  $[t_{нач}, t_{кон}]$ , которое приведет начальную точку  $x(t_{нач})$  в конечную  $x(t_{кон})$ .

Для понимания свойства управляемости, рассмотрим еще один пример. Имея математическое описание стационарного ОУ, сформулируем свойство управляемости, выведенное одним из известных ученых области современной теории управления – Рудольфом Калманом.

Согласно критерию Калмана, для стационарных объектов управления вида (2) существует матрица управляемости  $M_{упр} = [B, AB, A^2B, \dots, A^{n-1}B]$ , если ранг матрицы управления (3) будет равен порядку системы уравнений, описывающих ОУ, то рассматриваемый объект является полностью управляемым.

$$\frac{dx}{dt} = Ax + Bu, \quad x = \text{col}(x_1, \dots, x_n), \quad (2)$$

$$u = \text{col}(u_1, \dots, u_m),$$

$$\text{rank}(M_{упр}) = n \quad (3)$$

Описанный выше критерий является основополагающим принципом при исследовании объектов управления, поскольку свойство управляемости необходимо, в противном случае, теряется смысл синтеза закона управления.

### 1.5 Наблюдаемость

Современная теория управления позволяет автоматизировать процессы, протекающие в сложных объектах управления, не имеющих точное математическое описание. Одним из распространенных видов ОУ являются объекты, у которых возможно измерить только выходной сигнал. В этом случае становится невозможным измерение всех переменных состояния.

Для преодоления неопределенности, ввиду недостаточности данных о внутреннем состоянии ОУ, используются специальные устройства – наблюдатели и динамические корректоры. Но сложность применения таких устройств заключалась в том, что не для каждого объекта, по выходному сигналу возможно восстановить вектор переменных состояния. Ввиду этих ограничений, было сформулировано свойство восстанавливаемости состояний у ОУ. Данное свой-

ство показывало, возможно ли определить вектор состояний  $x(t)$  по будущим значениям выходного сигнала  $y(t)$ .

Сформулируем свойство восстанавливаемости для следующего ОУ:

$$\begin{aligned} \frac{dx}{dt} &= f(x, u, t), x = \text{col}(x_1, \dots, x_n), \\ y &= F(x, u, t), y = \text{col}(y_1, \dots, y_m), \end{aligned} \quad (4)$$

ОУ (4) будет называться полностью восстанавливаемым, при существовании некоторого  $s$ , которое принадлежит временному интервалу (5):

$$-\infty < s < t, \quad (5)$$

При этом, согласно измерениям выходного сигнала  $y(t_*)$  и управляющего сигнала  $u(t_*)$ , для всех  $t_*$ , входящих в интервал:

$$s < t_* < t, \quad (6)$$

становится возможным определить  $x(t)$ .

Если ОУ является стационарным и линейным, т. е. параметры такого ОУ не зависят от временного интервала, два понятия наблюдаемости и восстанавливаемости возможно объединить. В этом случае был сформулирован следующий критерий наблюдаемости, позволяющий доказать, что у ОУ возможно измерить вектор переменных состояний, имея только выходной сигнал.

Для ввода понятия наблюдаемости введем математическое описание стационарно ОУ:

$$\begin{aligned} \frac{dx}{dt} &= Ax + Bu, x = \text{col}(x_1, \dots, x_n), \\ y &= C^T x + Du, y = \text{col}(y_1, \dots, y_m), \end{aligned} \quad (7)$$

В этом случае, объект управления (7) является полностью наблюдаемым и восстанавливаемым, тогда и только тогда, когда ранг матрицы наблюдаемости  $M_{\text{набл}} = [C, A^T C, (A^T)^2 C, \dots, (A^T)^{n-1} C]$  равен порядку системы уравнений ОУ (8).

$$\text{rank}(M_{\text{набл}}) = n. \quad (8)$$

Для нестационарных нелинейных ОУ свойство наблюдаемости не зависит от ранга матрицы наблюдаемости. При рассмотрении сложных нелинейных нестационарных ОУ полностью наблюдаемым будет являться только тот объект,

для которого существует такое  $s$ , которое будет удовлетворять следующему условию:

$$t < s < +\infty, \quad (9)$$

таких, что согласно измерению выходного сигнала  $y(t_*)$  и управляющего воздействия  $u(t_*)$ , характерного для всех  $t_*$ , включенных в интервал:

$$t < t_* < s, \quad (10)$$

возможно восстановить вектор переменных состояния  $x(t)$ .

Как упоминалось ранее, существует несколько устройств, позволяющих измерять или восстанавливать вектор переменных состояния по выходному сигналу. Рассмотрим подробнее каждое из устройств.

Наблюдатель – устройство, используемое в паре с ОУ, позволяющее восстановить вектор переменных состояния по выходу объекта управления. Рассмотрим принцип работы устройства. Подавляющее большинство ОУ позволяют измерять выходной сигнал, поэтому при добавлении в контур адаптации наблюдателя, который устанавливается последовательно за объектом с недоступным вектором переменных состояния, формируется новый видоизмененный объект.

При этом, полученный видоизмененный объект позволяет измерить оценки вектора переменных состояния. Применение рассматриваемого метода является эффективным, поскольку измеряемые оценки недоступного вектора переменных состояния практически совпадает с реальными значениями исходного объекта.

Таким образом, становится возможным сформировать требуемую динамику у объекта без измерения реального вектора переменных состояния.

Существуют различные виды наблюдателей в зависимости от относительного порядка. В случае, если порядок наблюдателя равен порядку системы, устройство будет называться наблюдателем полного порядка. Данный наблюдатель формирует оценки всего вектора переменных состояния.

При понижении порядка наблюдателя на единицу, относительно объекта управления, получится устройство, позволяющее сформировать новый вектор

переменных состояния, учитывающий только недоступные измерению переменные состояния объекта.

Рассмотрим следующее устройство, позволяющее получить оценки недоступного вектора переменных состояния – динамический корректор. Данное устройство состоит из упругих звеньев в количестве  $k=(p-1)$  ( $p$  – относительный порядок передаточной функции ОУ) последовательно соединенных друг за другом.

При решении задачи в рамках данной магистерской диссертации было принято решение использовать динамический корректор или фильтр-корректор, математическое описание которого принимает вид:

$$u(s) = \frac{\delta(s)}{\gamma(s)} = \frac{\delta_0 s^{p-1} + \delta_1 s + 1}{(T_* s + 1)^{p-1}}, \quad (11)$$

где  $\delta(s)$  – гурвицев многочлен;

$\delta_i$  – положительные константы;

$T_* > 0$  – рассчитываемый числовой параметр;

$p$  – относительный порядок передаточной функции ОУ.

### **1.6 Методы управления объектами в условиях неопределенности**

Сложность структуры объектов управления растет с каждым годом, требуется управлять огромными системами, в состав которых может входить не один десяток сложных объектов управления, содержащих, как и неопределенность различного вида, так и всевозможные возмущающие воздействия, запаздывание, различные нелинейные и нестационарные динамические и статические характеристики процессов, протекающих внутри системы [13].

Для управления сложными объектами классические методы теории управления применять неэффективно. Для решения сложных задач теории управления были разработаны методы адаптивного и робастного управления объектами.

Робастное управление – совокупность методов в теории управления, основной целью которых является синтез требуемого регулятора, обеспечивающего заданное качество управления объектом. Данный метод управления приме-

няется при отсутствии математической модели объекта управления, поскольку в результате синтеза регулятором формируется такое воздействие, которое приводит выход объекта к желаемой динамики, применяется метод грубого воздействия.

Другим, в корне отличающимся от робастного, методом управления в условиях математической неопределенности является адаптивное управления. Адаптивные системы или самонастраивающиеся системы – это системы управления, которые обеспечивают некую компенсацию параметрических, сигнальных, функциональных или структурных неопределенностей объекта управления за счет автоматической подстройки адаптивного регулятора в ходе функционирования системы. Другими словами, априорная неопределенность восполняется информацией, получаемой в результате функционирования контура адаптации.

На рисунке 2 представлена классификация робастных и адаптивных систем управления.

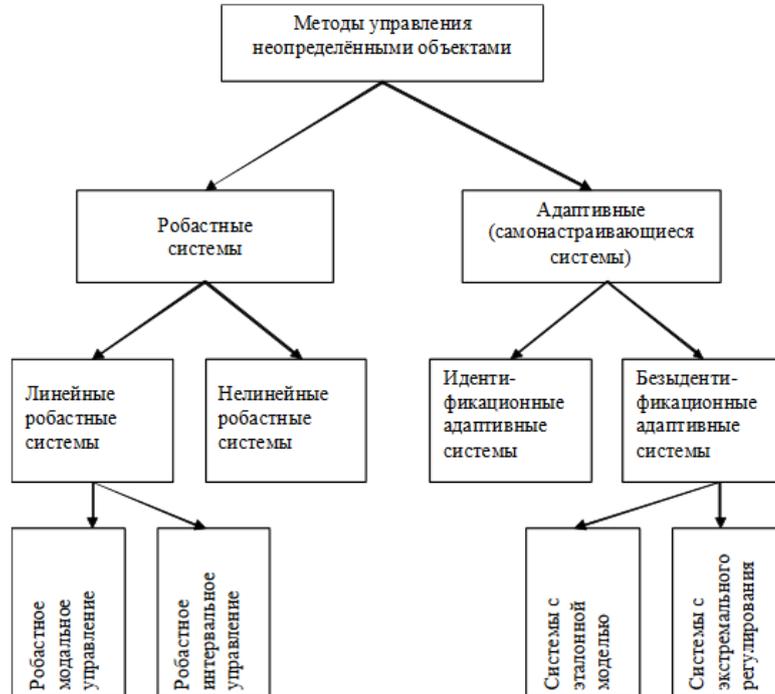


Рисунок 2 – методы управления в условиях неопределенности

Согласно рисунку 2, линейные робастные системы используют различные виды управления в условиях параметрической неопределенности, а именно: робастное модальное управления и робастное интегральное управления. Рассмотр-

ренные алгоритмы можно отнести неадаптивному классу, использующему векторно-матричное представление объекта управления, параметрическая неопределенность которого заключена в неопределенности матрицы состояния.

При использовании интервального робастного управления матрица состояний декомпозируется на интервальную и медианную составляющие, которые в свою очередь характеризуются свойством угловой реализации с одинаковыми нормами.

При использовании робастного управления нелинейными объектами управления с неопределенной математической моделью, используется искусственно вводимая в алгоритм управления статическая нелинейная обратная связь. Таким образом, данный алгоритм имеет малую чувствительность к неопределенной математической модели объекта.

Рассмотрим класс адаптивных систем. На рисунке 2 идентификационные адаптивные системы основаны на использовании процедуры идентификации объекта, в частности получении оценок параметров объекта управления в процессе функционирования системы. Полученные в результате функционирования оценки используются контуром адаптации для расчета коэффициентов регулятора, при этом контур адаптации в своей структуре будет иметь блок идентификации оценок, блок расчета требуемых параметров регулятора и автоматически настраиваемый регулятор. Основным недостатком применения текущего метода для получения желаемой динамики ОУ является задержки времени для получения оценок переменных состояния и выработке требуемого сигнала управления.

Безидентификационный метод адаптивного управления применяется для решения проблемы, связанной с потерей времени на настройку коэффициентов в момент идентификации. Принципом данного метода является настройка параметров регулятора по минимизации ошибки управления.

В таких системах цель управления задается с помощью эталонного значения регулируемой переменной (адаптивные безидентификационные системы с

эталонной моделью), либо с помощью некоторого вводимого искусственно числового критерия.

Рассмотрим принцип действия эталонной модели. Заданная разработчиком системы эталонная модель позволяет получить требуемый отклик на входное воздействие, таким образом становится возможным измерить отклонения ОУ от требуемой траектории. Получив сигнал рассогласования, адаптивный регулятор может выработать такое управляющее воздействие, что в замкнутой системе ОУ начнет стремиться к динамике эталона.

На данный момент существуют различные виды типовых эталонных моделей. При проектировании адаптивных систем управления используются: явные эталонные модели (ЯЭМ), неявные эталонные модели (НЭМ), а также комплексная эталонная модель, называемая явно-неявной эталонной моделью (ЯНЭМ), содержащая в себе как явную, так и неявную часть. Изучим принцип работы каждой из представленных эталонных моделей.

При рассмотрении системы управления с ЯЭМ, можно заметить, что в контур адаптации включается объект, задаваемый разработчиком, обладающий требуемой для технологического процесса динамикой. Текущая динамика ЭМ задает эталонное поведение, на основании которого адаптивный регулятор выработывает желаемое управление без потери времени на идентификации параметров.

НЭМ реализуется в форме уравнения, описывающего требуемое поведение системы для технологического процесса, таким образом в контур адаптации подстраивается к заданной разработчиком траектории движения.

Адаптивные системы с ЯНЭМ – системы, объединяющие два вида эталонных моделей, явную и неявную. Использование ЯНЭМ модели позволяет значительно упростить структуру системы управления, что является главным преимуществом использования систем с данной ЭМ.

Кроме того, основной особенностью применения эталонных моделей, при проектировании АСУ, является возможность изменять структуру устройства по отношению к другим элементам системы. Эталонная модель может включаться

в основной контур адаптации, при этом будет сформирован дополнительный контур с:

- параллельной эталонной моделью;
- последовательной эталонной моделью;
- с последовательно-параллельной системой управления.

На рисунке 3 показана структура дополнительного контура с параллельной эталонной моделью. Согласно рисунку 3 эталонная модель размещается параллельно контура самонастройки, что позволяет получить разность динамик между ЭМ и ОУ для формирования управляющего сигнала.

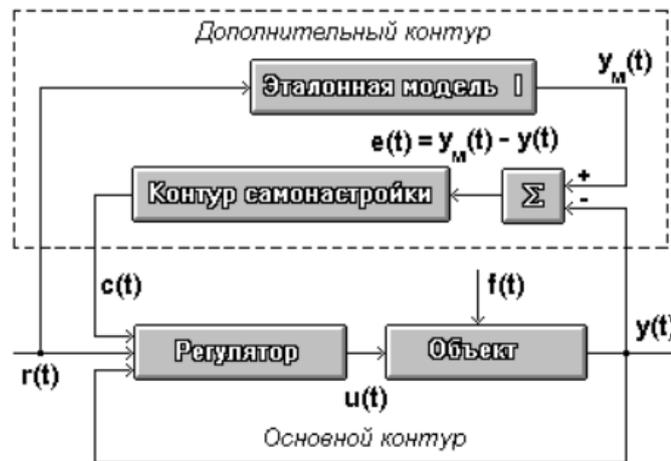


Рисунок 3 – Адаптивная система управления с параллельной ЭМ

Другим способом размещения эталонной модели в системе управления является установка последовательной ЭМ. На рисунке 4 представлена адаптивная система управления с последовательной эталонной моделью. В этом случае, контур самонастройки не использует выход ЭМ для формирования ошибки динамики, а эталонная модель формирует желаемое задающее воздействие, подающееся на вход регулятора.

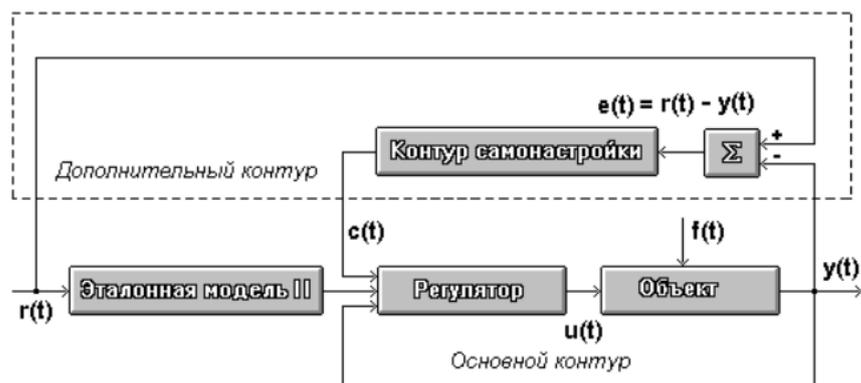


Рисунок 4 – Адаптивная система управления с последовательной ЭМ

Одним из самых эффективных методов построения СУ, использующей ЭМ является комбинация последовательной и параллельной ЭМ. Данный метод позволяет получить наиболее качественную ошибку отклонения, на основе которой адаптивный регулятор формирует желаемое поведение ОУ. На рисунке 5 показана структура ЯНЭМ модели, в этом случае контур самонастройки формирует желаемое поведение ОУ, используя сразу две эталонные модели.

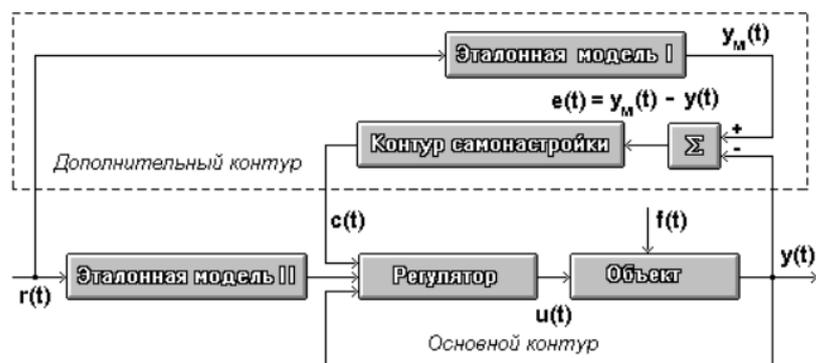


Рисунок 5 – Адаптивная система управления с последовательно-параллельной ЭМ.

Для формирования наиболее точного управляющего воздействия, в рамках выполняемой магистерской диссертации будет использована структура с ЯНЭМ.

### 1.7 Методы исследования устойчивости адаптивных систем

При построении и синтезе адаптивных систем управления, объект управления должен удовлетворять не только условию управляемости, но и условию устойчивости. Устойчивость – способностью объекта управления сохранять текущее состояние при наличии внешних воздействий окружающей среды, дан-

ное условия является так же основополагающим, поскольку систем, полностью изолированных от воздействия внешней среды – не существует. Для построения и синтеза устойчивых систем управления существуют различные методы исследования. В рамках исследования данной магистерской диссертации будут рассмотрены следующие методы исследования устойчивости ОУ:

- прямой метод Ляпунова;
- критерий гиперустойчивости Попова;
- метод абсолютной устойчивости Попова.

Прямой метод Ляпунова основан на формировании разработчиком специальных функций – функций Ляпунова. Данные функции рассматриваются в системе координат вида (12):

$$V(x_1, x_2, \dots, x_n) \tag{12}$$

Представленная выше функция обращается в нуль только в случае, когда все координаты становятся нулевыми. Кроме того, функция (12) является однозначной и непрерывной в области рассмотрения со всеми частными производными первого порядка. Для понимания применения функций Ляпунова, введем нелинейную систему со следующим математическим описанием:

$$\begin{cases} x_1 = t_1(x_1, x_2, \dots, x_n) \\ x_2 = t_2(x_1, x_2, \dots, x_n) \\ \text{-----} \\ x_n = t_n(x_1, x_2, \dots, x_n) \end{cases} \tag{13}$$

$$\frac{dV}{dt} = \frac{dV}{dx_1} * \frac{dx_1}{dt} + \frac{dV}{dx_2} * \frac{dx_2}{dt} + \dots + \frac{dV}{dx_n} * \frac{dx_n}{dt}, \tag{14}$$

Рассматривая заданную систему (12), возможно предположить, что для текущей системы существует такая функция, первая производная которой примет вид (14), а также будет отрицательно постоянной и отрицательно определенной, то этого будет достаточно для устойчивости системы. Данная функция будет называться функцией Ляпунова (12).

Другой метод исследования устойчивости системы – метод абсолютной устойчивости Попова позволяет установить устойчивость нелинейной системы сразу для целого класса нелинейности, лежащих в секторе.

Для подробного изучения метода устойчивости Попова, рассмотрим следующий рисунок:

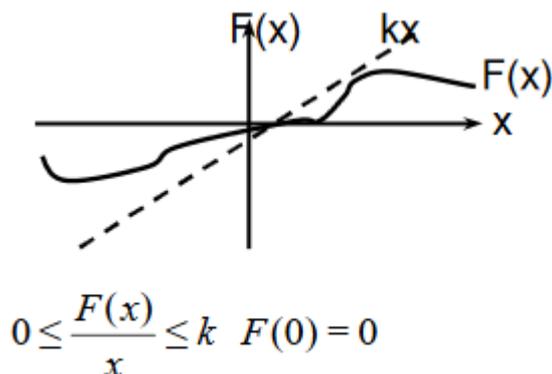


Рисунок 6 – нелинейность  $F(x)$  в секторе

Согласно рисунку 6, если нелинейность будет удовлетворять частотному условию, другими словами, нелинейность не будет выходить за рамки сектора 1 и 3 квадрантах, вид функции в этом случае не будет иметь значения, поскольку основным критерием является принадлежность рассматриваемому сектору.

Опираясь на рисунок 6, введем понятия устойчивости в целом и абсолютной устойчивости системы.

Устойчивость системы в целом заключается в устойчивости при любом начальном условии, т. е. объект не зависимо от точки начала движения должен прийти к желаемой динамике. Абсолютная устойчивость подразумевает устойчивость нелинейной системы управления в рамках заданного сектора. Другими словами, если разработчика будет устраивать границы устойчивости объекта согласно сектору, то для его системы ОУ можно считать абсолютно устойчивым.

Согласно введенным определениям, был сформулирован критерий абсолютной устойчивости Попова.

Пусть линейная часть ОУ имеет передаточную функцию  $W(p)$ , при этом, его нелинейная составляющая размещена в секторе  $k$ , тогда для устойчивости

ОУ в системе достаточно наличие такого Н, которое при любой частоте будет удовлетворять условию (15):

$$\operatorname{Re}[(1 + j\omega H) * W(j\omega)] > -\frac{1}{k} \quad (15)$$

Исследуя условия (15), возможно получить геометрическое представление устойчивости системы управления. Для этого развернем формулу (15) относительно скобок, тогда имеем:

$$\begin{aligned} \operatorname{Re}[(1 + jH\omega) * (\operatorname{Re}W + j \operatorname{Im}W)] &= \\ = \operatorname{Re}(\operatorname{Re}W + jH\omega \operatorname{Re}W + j \operatorname{Im}W - H\omega \operatorname{Im}W) &= \\ = \operatorname{Re}W - H\omega \operatorname{Im}W \end{aligned} \quad (16)$$

Другими словами, полученное выражение (16) показывает что:

$$\operatorname{Re}W - H\omega \operatorname{Im}W + \frac{1}{k} > 0 \quad (17)$$

Таким образом, возможно построить годограф (18), в силу которого, получим частотное неравенство (19):

$$\overline{W}(j\omega) = \operatorname{Re}W(j\omega) + j\omega \operatorname{Im}W(j\omega), \quad (18)$$

$$\operatorname{Re}\overline{W}(j\omega) + \frac{1}{k} > H \operatorname{Im}\overline{W}(j\omega) \quad (19)$$

Полученное таким образом условие (19) означает, что модифицированный годограф должен находиться правее прямой, проходящей через точку  $(-1/k; j0)$  с угловым коэффициентом Н. Полученные годографы представлены на рисунке 7.

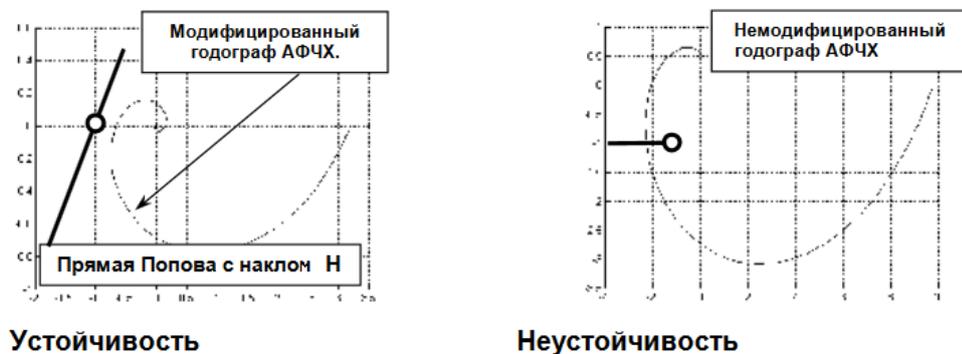


Рисунок 7 – Устойчивость и неустойчивость системы

Согласно рисунку 7, на левой части изображена устойчивая система, поскольку выполнены достаточные условия устойчивости. На правой части ри-

сунка 7 представлен годограф неустойчивой системы управления, поскольку необходимые и достаточные условия устойчивости модифицированного годографа не были выполнены.

Следующим критерием исследования устойчивости системы является критерий гиперустойчивости Попова. Основным отличием данного критерия от критерия абсолютной устойчивости попова является условие применения.

Если критерий абсолютной устойчивости Попова используется только для систем, удовлетворяющих условию (20), то критерий гиперустойчивости Попова имеет более широкое условие применения (21), что является несомненным преимуществом [7-8].

$$y(t) * f(y(t)) > 0, \quad (20)$$

$$\int_0^t y(v) * f(y(v)) dv \geq -\eta_0^2 = const, \forall t > 0. \quad (21)$$

Условие (21) более общее, поскольку интеграл больше константы в среднем.

Применение критерия гиперустойчивости Попова предполагает приведение исследуемой нелинейной системы управления к нелинейной системе первого типа, представленного на рисунке 8.

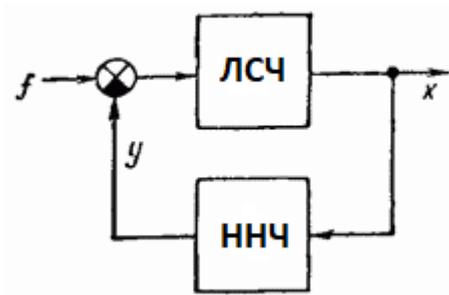


Рисунок 8 – Нелинейная система 1-го типа

Согласно рисунку 8, у исследуемой системы управления требуется выделить линейную стационарную часть (ЛСЧ) и нелинейную нестационарную часть (ННЧ).

В случае выполнения для системы следующих неравенств:

$$\text{Re}[W(jw)] > 0, \forall w > 0, \quad (22)$$

$$\int_0^t y(v) f(y(v)) dv \geq -\eta_0^2, \forall t > 0, \quad (23)$$

при упрощении принимающих следующий вид:

$$\eta(0, t) = -\int_0^t u(v) * y(u) dv \geq -\eta_0^2 = const, \forall t > 0, \quad (24)$$

то систему управления возможно считать гиперустойчивой.

Критерий гиперустойчивости Попова применим для синтеза систем адаптивного и робастного управления.

Главным достоинством применения рассматриваемого метода является возможность получить в явном виде алгоритмы настройки адаптивного регулятора при решении интегрального неравенства Попова (24), что в несколько раз упрощает построение требуемой системы управления.

Таким образом, после изучения существующих методов синтеза систем адаптивного управления,

### **1.8 Этапы синтеза систем по критерию гиперустойчивости Попова**

Критерий гиперустойчивости Попова, применяемый для исследования устойчивости и синтеза системы управления основывается на 4 этапах синтеза адаптивных систем. Для синтеза системы управления без потери времени на определение параметров будет использован беспойсковый метод, позволяющий получить желаемую динамику без задержек во времени.

Согласно четко определенному алгоритму синтеза системы на основе критерия гиперустойчивости Попова, на первом этапе требуется определить эквивалентное математическое описание системы 1-го типа, описанного на рисунке 8, т. е. получить линейную стационарную часть (ЛСЧ) и нелинейную нестационарную часть (ННЧ) представленную в виде:

$$\begin{aligned} \frac{dx}{dt} &= Ae(t) + Bq(t), z_0(t) = g^T L^T e(t), \\ q(t) &= -F(z_0(t)). \end{aligned} \quad (25)$$

Второй этап синтеза на основе критерия гиперустойчивости Попова заключается в решении интегрального неравенства Попова (ИНП) и получении

настроек адаптивного регулятора в явном виде, ИНП принимает следующий вид:

$$h(0, t) = -\int_0^t q^T(s) z_0(s) ds = \int_0^t F^T(z_0(s)) z_0(s) ds > -h_0, \quad (26)$$

Следующий этап синтеза адаптивной системы управления на основе критерия гиперустойчивости Попова заключается в определении в условиях априорной неопределенности числовых значений вектора  $\mathbf{g}$ . Главной особенностью данного этапа является то, что выбор числовых значений вектора  $\mathbf{g}$  должно обеспечивать положительность линейной стационарной части (ЛСЧ) системы первого типа.

Поскольку ЛСЧ и ННЧ являются независимыми друг от друга, то очередность проведения этапов синтеза может быть изменена, данный факт не повлияет на результат синтеза на основе рассматриваемого критерия. Все условия априорной неопределенности должны быть сформулированы до синтеза на основе критерия гиперустойчивости Попова, поскольку данный факт является важным при разрешении интегрального неравенства Попова.

Последним этапом синтеза адаптивной системы управления на основе критерия гиперустойчивости Попова является проверка полученных решений в соответствии поставленным целям и задачам управления.

Таким образом, были определены и описаны все этапы синтеза требуемой адаптивной системы управления на основе критерия гиперустойчивости Попова.

## 2 СИНТЕЗ АДАПТИВНОЙ СИСТЕМЫ УПРАВЛЕНИЯ

### 2.1 Математическое описание объекта управления

Для разработки и построения любой системы управления требуется знать точное математическое описание всех ее элементов для формирования системы дифференциальных уравнений. Зададим точное математическое описание всех компонентов системы.

Зададим математическое описание объекта управления:

$$\begin{aligned} \frac{d\mathbf{x}^{(q)}(t)}{dt} &= \mathbf{A}_q \mathbf{x}^{(q)}(t) + \mathbf{D} \mathbf{x}^{(q)}(t - \tau) + \mathbf{B} u^{(q)}(t), y^{(q)}(t) = x^{(q)}_1(t), \\ x^{(q)}(h) &= F(h), h \in [-\tau; 0], q = 1, 2, \dots, Q, \end{aligned} \quad (27)$$

где  $q = 1, 2, \dots, Q$  – ограниченное количество интервалов времени;

$x^{(q)T}(t) = [x_1^{(q)}(t) \ x_2^{(q)}(t) \ x_3^{(q)}(t)]$  – вектор состояния на интервале времени  $q$ ;

$\tau = const > 0$  – известное временное запаздывание переменных состояния;

$x^{(q)}(h)$  – непрерывная и ограниченная во времени начальная функция;

$u^{(q)}(t)$  – формируемое скалярное управление;

$y^{(q)}(t)$  – вектор выхода;

$\mathbf{B}$  – вектор управления;

$\mathbf{A}_q$  и  $\mathbf{D}$  матрицы заданного размера  $n \times n$  заданные в форме:

$$\mathbf{A}_q = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ a_{1q} & a_{2q} & a_{3q} \end{pmatrix}, \mathbf{B} = \begin{pmatrix} 0 \\ 0 \\ b_1 \end{pmatrix}, \mathbf{D} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ d_1 & d_2 & d_3 \end{pmatrix}, \quad (28)$$

Описанный объект управления (27) функционирует в условиях априорной неопределенности, представленной в виде:

$$\mathbf{A}_q = \mathbf{A}_q(\xi), \mathbf{B} = \mathbf{B}(\xi), \mathbf{D} = \mathbf{D}(\xi), \xi \in \Xi, \quad (29)$$

где  $\xi$  – набор неопределенных параметров множества  $\Xi$ . При определении явного вида априорной неопределенности объекта управления (27), параметры задаются в виде:

$$\begin{aligned} \min a_{1q} < a_{1q} < \max a_{1q}, \min a_{2q} < a_{2q} < \max a_{2q}, \min a_{3q} < a_{3q} < \max a_{3q} \\ \min d_1 < d_1 < \max d_1, \min d_2 < d_2 < \max d_2, \min d_3 < d_3 < \max d_3, \\ \min b_1 < b_1 < \max b_1 \end{aligned} \quad (30)$$

Основной особенностью рассматриваемого объекта управления, функционирующего в условиях априорной неопределенности, является скачкообразное изменения динамики на каждом из  $q$  интервалах времени [12].

Уравнению (27) соответствует передаточная функция на интервале времени  $q$ , представленная в виде:

$$W_{OY_q}(s) = \frac{\alpha_q(s)}{\beta_q(s)} = \frac{1}{s^p + \beta_{11q}s^{p-1} + \dots + \beta_{p1q} + \sum_{l=2}^{\theta} (\beta_{l1q}s^{p-1} + \dots + \beta_{plq})e^{-s\tau}}, \quad (31)$$

где  $W_{OY_q}(s)$  – передаточная функция объекта управления (ОУ);

$\alpha_q(s)$  – гурвицев полином;

$\beta_q(s)$  – квазиполином с произвольным расположением корней;

$p$  – степень квазиполинома.

Главной целью при разработке систем управления является придание объекту управления желаемой динамики. При проектировании в частности адаптивных систем управления, требуется задать эталонную модель, которая будет формировать динамику, к которой должен стремиться объект. Поскольку эталонная модель вводится в контур управления искусственно, то она имеет полное описание всех элементов и не имеет неопределенностей. Зададим математическое описание эталонной модели в виде:

$$\frac{dx_M(t)}{dt} = \mathbf{A}_M x_M(t) + \mathbf{B}_M r(t), \quad y_M(t) = x_{M1}(t), \quad (32)$$

где  $r(t)$  – задающее воздействие; структуры матриц  $\mathbf{A}_M, \mathbf{B}_M$  имеют вид:

$$\mathbf{A}_M = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ a_{1M} & a_{2M} & a_{3M} \end{pmatrix}, \quad \mathbf{B}_M = \begin{pmatrix} 0 \\ 0 \\ b_{1M} \end{pmatrix}, \quad (33)$$

Согласно математическому описанию объекта управления (27) для каждого  $q$ -го интервала времени, по выходу возможно измерить только переменную состояния  $x_{q1}$ . Для придания ОУ желаемой динамики, заданной эталонной моделью требуется полностью измеряемый вектор переменных состояния. В данном случае, для получения оценок всех переменных состояния используется

динамический корректор. Подробное описание применения динамического корректора представлено в приложении А.

При применении динамического корректора обеспечивается L-диссипативность упрощенной системы управления, кроме того, становятся доступными оценки вектора переменных состояния, что позволяет получить следующее математическое описание уже видоизмененного объекта управления (ВОУ):

$$\begin{aligned} \frac{d\bar{x}^{(q)}(t)}{dt} &= \mathbf{A}_q \bar{\mathbf{x}}^{(q)}(t) + \mathbf{D}\bar{\mathbf{x}}^{(q)}(t-\tau) + \mathbf{B}u^{(q)}(t), y^{(q)}(t) = \bar{x}_1^{(q)}(t), \\ z^{(q)}(t) &= \mathbf{g}^T \bar{\mathbf{x}}^{(q)}(t), \bar{x}^{(q)}(h) = F(h), q \in [-\tau; 0], q = 1, 2, \dots, Q, \end{aligned} \quad (34)$$

где  $\bar{\mathbf{x}}$  – оценки вектора пространства состояний;

$y(t)$  – выход объекта управления;

$z(t)$  – обобщенный выход объекта управления;

$\mathbf{g}^T$  – вектор выхода, формируемый из коэффициентов  $g_i$ , вычисляемых из

соотношения:

$$a^{(q)}(s)\delta(s) = (g_1 s^{p-1} + \dots + g_{p-2} s + g_{p-1}), \quad (35)$$

где  $a^{(q)}(s)\delta(s)$  – числитель передаточной функции ВОУ [1].

При наличии в полученном видоизмененном объекте управления (34) обобщенного выхода  $z(t)$ , требуется задать обобщенный выход эталона, таким образом, дополним эталонную модель вида (32) с обобщенным выходом получим следующее математическое описание ЭМ:

$$\begin{aligned} \frac{dx_M(t)}{dt} &= \mathbf{A}_M x_M(t) + \mathbf{B}_M r(t), y_M(t) = x_{M1}(t), \\ Z_M(t) &= \mathbf{g}^T \mathbf{x}_M \end{aligned} \quad (36)$$

Согласно выбранной структуре адаптивного регулятора существуют постоянные векторы  $\mathbf{C}_1$ ,  $\mathbf{C}_2$  и число  $K_0$ , что для объекта управления (34) и эталонной модели (36) будут выполняться условия структурного согласования:

$$\mathbf{A}_M - \mathbf{A}^{(q)} = \mathbf{B}^{(q)} \mathbf{C}_1^{(q)T}, \mathbf{B}_M = \mathbf{B}^{(q)} K_0^{(q)}, \mathbf{D}^{(q)} = \mathbf{B}^{(q)} \mathbf{C}_2^{(q)T}, \quad (37)$$

Условия структурного согласования показывают, что объект управления структурно согласован с эталонной моделью, поскольку, только в этом случае возможно ввести ошибку вида:

$$e(t) = x_M(t) - \overline{x^{(q)}}(t), \quad (38)$$

Таким образом, было сформировано математическое описание всех элементов адаптивной системы управления, при этом задано условие структурного согласования, получено эквивалентное математическое описание объекта управления, позволяющее применить критерий гиперустойчивости Попова для синтеза требуемой адаптивной системы.

## 2.2 Постановка задачи

Постановка задачи управления является главной составляющей синтеза требуемой адаптивной системы управления, поскольку, на этом этапе происходит определение необходимого качества функционирования синтезированной системы в целом. В рамках выполнения магистерской диссертации была задана следующая постановка задачи синтеза адаптивной системы управления объектом с запаздыванием на множестве состояний функционирования.

Для рассматриваемого ОУ, представленного в виде вида (34), функционирующего в условиях априорной неопределенности требуется построить замкнутую систему управления с помощью адаптивного регулятора, следующей структуры:

$$u^{(q)}(t) = K^{(q)}(t)r(t) + C_1^{(q)T}(t)\overline{x^{(q)}}(t) + C_2^{(q)T}(t)\overline{x^{(q)}}(t-\tau), \quad (39)$$

где  $K^{(q)}(t)$  – настраиваемый в процессе функционирования коэффициент по задающему воздействию, а  $C_1^{(q)}(t)$  и  $C_2^{(q)}(t)$  векторы определяемых в результате работы контура адаптации коэффициентов. В явном виде точное математическое описание элементов адаптивного регулятора имеет вид:

$$\begin{aligned} K^{(q)}(t) &= F_1(\overline{x^{(q)}}(t), x_M(t), r(t), t), \\ C_1^{(q)}(t) &= F_2(\overline{x^{(q)}}(t), x_M(t), t), \\ C_2^{(q)}(t) &= F_3(\overline{x^{(q)}}(t-\tau), x_M(t), t), \end{aligned} \quad (40)$$

Настройки всех коэффициентов требуется получить в результате синтеза адаптивной системы управления согласно критерию гиперустойчивости Попова. Задачу синтеза на основе критерия гиперустойчивости Попова необходимо решить таким образом, чтобы реакция заданной системы (34), (36), (38), (39) на задающее воздействие  $r(t)$ , по окончании процесса адаптации совпадала бы с динамикой выбранной эталонной моделью вида (36). По окончании процесса адаптации, система должна удовлетворять следующим целевым условиям:

$$\begin{aligned}\lim_{t \rightarrow \infty} e(t) &= \lim_{t \rightarrow \infty} (x_M(t) - x^{-(q)}(t)) = 0, \\ \lim_{t \rightarrow \infty} e(t) &= \lim_{t \rightarrow \infty} (y_M(t) - y^{(q)}(t)) = 0,\end{aligned}\tag{41}$$

и условий адаптации:

$$\begin{aligned}\lim_{t \rightarrow \infty} |C^{(q)}_1(t)| &\leq C^{(q)}_{01i}, C^{(q)}_{01} = const > 0, \\ \lim_{t \rightarrow \infty} |C^{(q)}_2(t)| &\leq C^{(q)}_{02i}, C^{(q)}_{02} = const > 0, \\ \lim_{t \rightarrow \infty} |K^{(q)}(t)| &\leq K^{(q)}_0, K^{(q)}_0 = const > 0.\end{aligned}\tag{42}$$

В случае достижения ВОУ условий, заданных уравнениями (41) и (42) можно сделать вывод, что и ОУ (27) в связке с ЭМ (32) также будет соответствовать всем заданным целевым условиям и условиям адаптации, поскольку в результате математического описания ВОУ было доказано, что ВДК в виду его малого влияния на систему, возможно рассматривать как внешнее возмущение среды.

### 2.3 Синтез алгоритмов системы адаптации

Поскольку в результате исследования методов синтеза адаптивных систем управления был выбран критерий гиперустойчивости Попова, то в рамках текущей магистерской диссертации синтез будет производиться на основе типовой последовательности этапов синтеза.

Первый этап синтеза на основе критерия гиперустойчивости Попова предполагает получение эквивалентного математического описание 1-го типа (ЭМО), представленное на рисунке 8. Для выделения нелинейной нестационарной части (ННЧ) из общего уравнения синтезированной системы управления,

примем ошибку в виде (38), в результате чего, получим следующее математическое описание системы:

$$\frac{de(t)}{dt} = \mathbf{A}_M e(t) + \mathbf{B}\mu(t), \quad z_0 = \mathbf{g}^T (z_M(t) - z^{(q)}(t)) = \mathbf{g}^T e(t), \quad (43)$$

$$\mu = -(\mathbf{K}(t) - \mathbf{K}^{(q)}_0)r(t) - (\mathbf{C}^{(q)}_{1i}(t) - \mathbf{C}^{(q)}_{01i})^T \bar{\mathbf{x}}^{(q)}(t) - (\mathbf{C}^{(q)}_{2i}(t) - \mathbf{C}^{(q)}_{02i})^T \bar{\mathbf{x}}^{(q)}(t - \tau), \quad (44)$$

где (43) – линейная стационарная часть, (44) – нелинейная нестационарная часть.

Второй этап синтеза на основе критерия гиперустойчивости предполагает разрешение интегрального неравенства Попова (ИНП) относительно выведенной ННЧ. Сформированное ИНП для рассматриваемой системы представлен в виде:

$$\eta(0, t) = -\int_0^t \mu(s) z_0(s) ds \geq -\gamma_0^2 = const, \quad \forall t > 0, \quad (45)$$

Для разрешения полученного ИНП используем метод суммы интегралов, для упрощения системы, полученная система интегралов показана ниже:

$$\begin{aligned} \eta(0, t) = & \int_0^t (K(s) - K^{(q)}_0)r(s)\mathbf{g}^T e(s)ds + \\ & + \sum_{i=1}^3 \int_0^t [(\mathbf{C}^{(q)}_{1i} - \mathbf{C}^{(q)}_{01i})^T \bar{\mathbf{x}}^{(q)}(s)]\mathbf{g}^T e(s)ds + \\ & + \sum_{i=1}^3 \int_0^t [(\mathbf{C}^{(q)}_{2i} - \mathbf{C}^{(q)}_{02i})^T \bar{\mathbf{x}}^{(q)}(s - \tau)]e(s), \end{aligned} \quad (46)$$

В ходе разрешения интегрального неравенства Попова (ИНП) было принято решение задать алгоритмы настройки параметров адаптивного регулятора в следующем виде:

$$\begin{aligned} K^{(q)}(t) &= \alpha \int_0^t \mathbf{g}^T e(s)r(s)ds + K^{(q)}(0), \\ C^{(q)}_{1i} &= \beta_i \int_0^t \mathbf{g}^T e(s)\bar{\mathbf{x}}^{(q)}(s)ds + C^{(q)}_{01i}(0), \\ C^{(q)}_{2i} &= \theta_i \int_0^t \mathbf{g}^T e(s)\bar{\mathbf{x}}^{(q)}(s - \tau)ds + C^{(q)}_{02i}(0). \end{aligned} \quad (47)$$

где  $\alpha$ ,  $\beta_i$  и  $\theta_i$  – соответствующие числа;

$i=1,2,3$ ; тогда с учетом настроек (47), обеспечим выполнение ИНП:

$$\begin{aligned}
h(0,t) &\geq -\frac{1}{2}\alpha^{-1}\int_0^t (K^{(q)}(0) - K^{(q)_0})^2 - \\
&-\frac{1}{2}\sum_{i=1}^3 \beta_i^{-1} (C^{(q)}_{01i}(0) - C^{(q)}_{1i})^2 - \\
&-\frac{1}{2}\sum_{i=1}^3 \theta_i^{-1} (C^{(q)}_{02i}(0) - C^{(q)}_{2i})^2 = -h_0 = const < 0,
\end{aligned} \tag{48}$$

В результате разрешения ИНП, были получены настройки адаптивного регулятора в явном виде для интервала времени  $q=1=Q$ . Поскольку рассматриваемый объект управления имеет сложную структуру, а именно,  $(q>1)=Q$  состояний функционирования на всем временном интервале, при этом, изменение динамики ОУ происходит скачкообразно. В этом случае, в контур адаптации вводится дополнительное устройство, позволяющее погасить резко возросшую ошибку при смене динамики ОУ. Введенное устройство представляет из себя зону нечувствительности, которая преобразует алгоритмы настройки адаптивного регулятора, полученные на этапе разрешения ИНП в вид, представленный ниже:

$$\begin{aligned}
\frac{dC^{(q)}_1(t)}{dt} &= \begin{cases} K^{(q)}(z_0^{(q)}(t) - z^{(q)}(t))r(t), \forall |z_0^{(q)}(t) - z^{(q)}(t)| > \Delta \\ 0, \forall |z_0^{(q)}(t) - z^{(q)}(t)| < \Delta \end{cases} \\
\frac{dC^{(q)}_2(t)}{dt} &= \begin{cases} K^{(q)}(z_0^{(q)}(t) - z^{(q)}(t))r(t), \forall |z_0^{(q)}(t) - z^{(q)}(t)| > \Delta \\ 0, \forall |z_0^{(q)}(t) - z^{(q)}(t)| < \Delta \end{cases},
\end{aligned} \tag{49}$$

где  $\Delta$  – малая величина, задаваемая пользователем системы.

Данная модификация позволяет исключить скачкообразный рост ошибки на момент адаптации системы к новым параметрам ВОУ.

Третий этап синтеза системы управления по критерию гиперустойчивости предполагает доказательство строгой положительности линейной стационарной части (ЛСЧ) в виде (43). Действительно, для упрощенной системы необходимо и достаточно выбрать значения вектора  $\mathbf{g}$  таким образом, чтобы выполнялось условие вещественности и положительности:

$$\operatorname{Re}(W_{\text{воу}}^{(q)}(jw)) > 0, \forall w \in (-\infty; +\infty) \tag{50}$$

Четвертый этап синтеза на основе критерия гиперустойчивости Попова заключается в проверке выполнимости поставленных целей адаптации, провер-

ку выполнимости возможно провести в результате получения имитационных характеристик системы. В рамках данной магистерской диссертации в следующей главе будет построена имитационная модель и получены требуемые характеристики системы для проверки работоспособности контура адаптации.

В результате проведения синтеза на основе критерия гиперустойчивости Попова, было получено эквивалентное математическое описание в виде ЛСЧ и ННЧ. Далее была доказана положительность ЛСЧ и разрешено ИНП, получены настройки адаптивного регулятора в явном виде. Также была доказана асимптотическая гиперустойчивость и L-диссипативность системы. Таким образом, имея все исходные данные для построения системы управления возможно приступить к имитационному моделированию синтезированной системы управления.

## 3 ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ

### 3.1 Описание среды моделирования

#### 3.1.1 Обзор сред моделирования

Работа с имитационной моделью помогает, во-первых, более полно изучить объект управления в условиях эксплуатации, поскольку на практике невозможно воссоздать все возможные ситуации, а во-вторых, имитационное моделирование дает возможность более эффективно выполнить исследование, проектирование и оптимизацию технологического процесса [10].

Для моделирования динамических систем существует множество программных комплексов, обладающих различным функционалом и сложностью моделирования. Рассмотрим наиболее распространенные программные пакеты, позволяющие провести оценку работоспособности синтезированной системы управления:

- Scilab & Xcos;
- Simulink;
- SimulationX.

Рассмотрим пакет имитационного моделирования Xcos, входящий в состав приложения системы Scilab. При моделировании с использованием Xcos реализуется принцип визуального программирования, в соответствии с которым разработчик на экране из стандартных блоков, входящих в библиотеку создает модель устройства и осуществляет расчеты.

Одним из главных достоинств данного пакета является простота моделирования, поскольку пользователю не требуется знать язык программирования и численные методы математики, стандартные блоки позволяют решить большинство стандартных задач. В случае нетипового способа решения задачи, пакет позволяет создавать собственные библиотеки блоков.

При моделировании имеется возможность следить за процессами, происходящими в системе с помощью специальных устройств наблюдения, входящих в состав библиотеки Xcos.

Для каждого блока в Xcos пользователь может настроить интересующие его параметры, переменные, либо настройки решения дифференциальных уравнений.

Simulink – это графическая среда имитационного моделирования входящая в состав мощного математического пакета MATLAB, позволяющая при помощи блок-диаграмм в виде направленных графов, строить динамические модели, включая дискретные, непрерывные и гибридные, нелинейные и разрывные системы [14].

Главным достоинством данного пакета является интеграция с ядром MATLAB, которое позволяет получить мощный инструмент для работы с матричным вычислением. Моделирование и настройка имитационной модели аналогично пакету Xcos.

Еще одним из распространенных пакетов моделирования является SimulationX. SimulationX – междисциплинарное многодоменное программное обеспечение для моделирования, анализа и оптимизации сложных систем на единой платформе. Более 500 элементов модели упрощают и ускоряют процесс моделирования.

Достоинством данного пакета является поддержка языка Modelica, который используется в частности для создания пользователем собственных моделей. Данная система позволяет моделировать переходные процессы динамических систем, но основной задачей данного продукта является моделирование, симулирование, анализ и виртуальное тестирование сложных мехатронных систем.

Таким образом, для имитационного моделирование синтезированной адаптивной системы управления будет использована среда MATLAB с пакетом моделирования Simulink, поскольку, мощное ядро матричного вычисления, визуальный интерфейс построения схемы, встроенные средства просмотра результатов моделирования в полной мере позволят произвести исследование, оценку качества производительности и оптимизацию всех элементов системы.

### 3.1.2 Среда моделирования MATLAB

Одним из самых известных и мощных пакетов прикладных программ является MATLAB – разработка американской компании, специализирующейся на разработке программного обеспечения для сложных матричных математических вычислений с поддержкой модулей имитационного моделирования – MathWorks. Данный пакет прикладных программ включает в состав огромный набор различных функций и специализированных библиотек, позволяющих использовать продукт в любых сферах, требующих математические вычисления различной сложности, визуализацию данных, разработку алгоритмов, пользовательских интерфейсов и специальных MEX-файлов. Кратко рассмотрим возможности пакета прикладных программ MATLAB.

Основной особенностью текущего продукта является мощная математическая база, позволяющая выполнять вычисления различной сложности. При помощи встроенных библиотек, возможно использовать статическую регрессию, применять преобразования Фурье, проводить интерполяцию и экстраполяцию кривых отрезков. Встроенные библиотеки MATLAB позволяют использовать векторно-матричное вычисление, разрешать линейные и нелинейные уравнения, выполнять поиск корней многочленов и операции над различными многочленами. Текущий поддерживаемый функционал позволяет разрешить практически любую расчетную задачу, что является безоговорочным преимуществом продукта над конкурентами.

Кроме того, MATLAB это не только набор заранее загруженных библиотек, позволяющих решать типовые задачи, он позволяет использовать высокоуровневый язык программирования, поддерживающий технологию объектно-ориентированности. Встроенный язык позволяет разрабатывать различного рода программное обеспечение на встроенной платформе MATLAB Compiler Runtime (MCR), а встроенный интерпретатор, профайлер и отладчик способствуют процессу разработки.

Так же MATLAB включает огромный набор встроенных функций, позволяющих работать с графическими данными, а именно: выполнять построение графиков, анализировать изображения, использовать трехмерную анимацию и т. д.

Поскольку пакет прикладных программ MATLAB позволяет разрабатывать независимое программное обеспечение, в состав MCR входит поддержка внешних интерфейсов. Данная поддержка позволит использовать технологию COM, что является несомненно одним из достоинств программного продукта, поскольку написанные на разных языках программирования приложения, смогут используя протоколы COM взаимодействовать со средой MATLAB.

Не менее важной составляющей данной среды является использование MEX-файлов, позволяющих использовать динамические библиотеки, разработанные для решения прикладной задачи. Кроме того, в MATLAB существует возможность использовать встроенные функции через C-интерфейс, что позволяет использовать функции пакета во внешних приложениях, написанных на языке программирования C, данная технология называется Engine.

### 3.1.3 Описание модуля Simulink

В рамках данной магистерской диссертации для имитационного моделирования использовалась встроенная в MATLAB графическая среда имитационного моделирования Simulink [14].

Simulink – это среда моделирования и проектирования на основе моделей для динамических и встроенных систем, интегрированная с MATLAB. Рассматриваемая среда моделирования была разработана компанией MathWorks и представляет из себя инструмент языка графического программирования потоков данных для моделирования, а также анализа многодоменных динамических систем. Основой данного продукта являются конструкторы блок-схем, с встроенными библиотеками блоков, позволяющими смоделировать большинство процессов физической, химической и электрической природы. Реализованный конструктор блок-схем позволяет включать алгоритмы системы MATLAB, а

также экспортировать результаты моделирования в общую среду MATLAB для дальнейшего анализа и обработки имитационных данных.

Simulink поддерживает большинство технологий:

- системный анализ данных;
- моделирования процессов;
- автоматическая генерация кода из конструктора блок-схем;
- тестирование и проверка встроенных систем.

Встроенные в модуль библиотеки блоков для моделирования систем различной природы позволяют моделировать процессы:

- тепловые;
- гидравлические;
- электрические;
- механические;
- электросиловые.

Использование встроенного конструктора блок-схем позволяет применять развитый модельно-ориентированный подход для разработки любых систем адаптивного управления. Одними из основополагающих особенностей графической среды являются:

- удобная интерактивная графическая среда для построения блок-диаграмм;
- встроенные средства, позволяющие отлаживать и анализировать построенные модели;
- полная интеграция с MATLAB, включая визуализацию данных, методы решения и анализ, а также, графические интерфейсы;
- встроенные современные средства решения дифференциальных уравнений для систем различного типа;
- возможность имитационного моделирования нестационарных систем с помощью встроенных решателей с задаваемым или переменным шагом;
- расширяемая библиотека готовых блоков.

Таким образом, рассмотренный модуль, входящий в состав пакета прикладных программ MATLAB полностью соответствует требованиям для построения имитационной модели синтезированной адаптивной системы управления с

запаздыванием на множестве состояний функционирования.

### 3.2 Задание параметров системы

Для рассматриваемого объекта управления (34), где коэффициенты передаточных функций  $W_{OY(q)}$  поэтапно принимают кусочно-постоянные значения на интервалах времени:

$$\begin{aligned}
 t_0 &= 0; \beta_{11} = -0.1; \beta_{21} = -0.2; \beta_{31} = -0.6; \\
 t_1 &= 50; \beta_{12} = 0.4; \beta_{22} = 2.1; \beta_{32} = 0.4; \\
 t_2 &= 75; \beta_{13} = 0.6; \beta_{23} = -2.9; \beta_{33} = 0.9; \\
 t_3 &= 200; \beta_{14} = 0.1; \beta_{24} = 1.7; \beta_{34} = 1.2; \\
 t_4 &= 400; \beta_{15} = 1.9; \beta_{25} = -1.5; \beta_{35} = -2.
 \end{aligned}
 \tag{51}$$

Матрица управления и запаздывания заданы в виде:

$$\mathbf{B} = \begin{pmatrix} 0 \\ 0 \\ 0.1 < b_1 < 2 \end{pmatrix},$$

$$\mathbf{D} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 7 & -4 & -2 \end{pmatrix}.$$
(52)

Для разрешения проблемы строгой положительности и вещественности передаточных функций объекта управления на всех интервалах времени, вектор  $\mathbf{g}$  был выбран следующим образом:

$$\mathbf{g} = \begin{pmatrix} 1.5 \\ 2.5 \\ 1 \end{pmatrix}.$$
(53)

Явно-неявная эталонная модель (36), задающая требуемую динамику для объекта управления, была задана в следующем виде:

$$\mathbf{A}_M = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 3 & 2.75 & 0.75 \end{pmatrix}, \mathbf{B}_M = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}. \quad (54)$$

В соответствии с реакцией объекта управления на задающее воздействие было задано  $r(t)$  такое, что инертность объекта управления минимизируется, для получения более точных данных о работе контура адаптации, таким образом, задающее воздействие принимает следующий вид:

$$r(t) = 0.4(1 - \cos((0.09t) - 1)\sin(0.02t)). \quad (55)$$

Поскольку ОУ был модифицирован динамическим корректором, то говорить о его устойчивости невозможно, но в виду принятия ВДК в виде внешнего возмущения, то возможно обеспечить условия L-диссипативности системы, что в рамках решения данной задачи приравнивается к гиперустойчивости системы, выбрав параметры динамического корректора в соответствии с алгоритмом расчета представленным в приложении А. Кроме того при использовании явно- неявной эталонной модели требуется чтобы полином  $\delta(s)$  был равен вектору  $\mathbf{g}$ , таким образом зададим динамический корректор в виде:

$$W_{ДК} = \frac{1.5\delta_1^2 + 2.5\delta_2 + 1}{(0.002s + 1)^2}. \quad (56)$$

Постоянное временное запаздывание по состоянию  $\tau=2$ сек.

Построенная Simulink схема синтезированной адаптивной системы объектом управления на множестве состояний функционирования представлена в приложении Б на рисунке Б.1.

При настройке адаптивного регулятора параметры коэффициентов были подобраны в следующем виде:

$$\begin{aligned} C_{11} &= 5000, C_{21} = 5000, \\ C_{12} &= 6000, C_{22} = 6000, \\ C_{13} &= 7500, C_{23} = 7000, \\ K_0 &= 8000. \end{aligned} \quad (57)$$

### 3.3 Характеристики системы

При моделировании синтезированной адаптивной системы управления были получены следующие характеристики.

На рисунке 9 представлена динамика изменения параметров объекта управления на каждом из выбранных интервалах времени. Согласно графикам, параметры изменяются скачкообразно в каждый из промежутков времени.

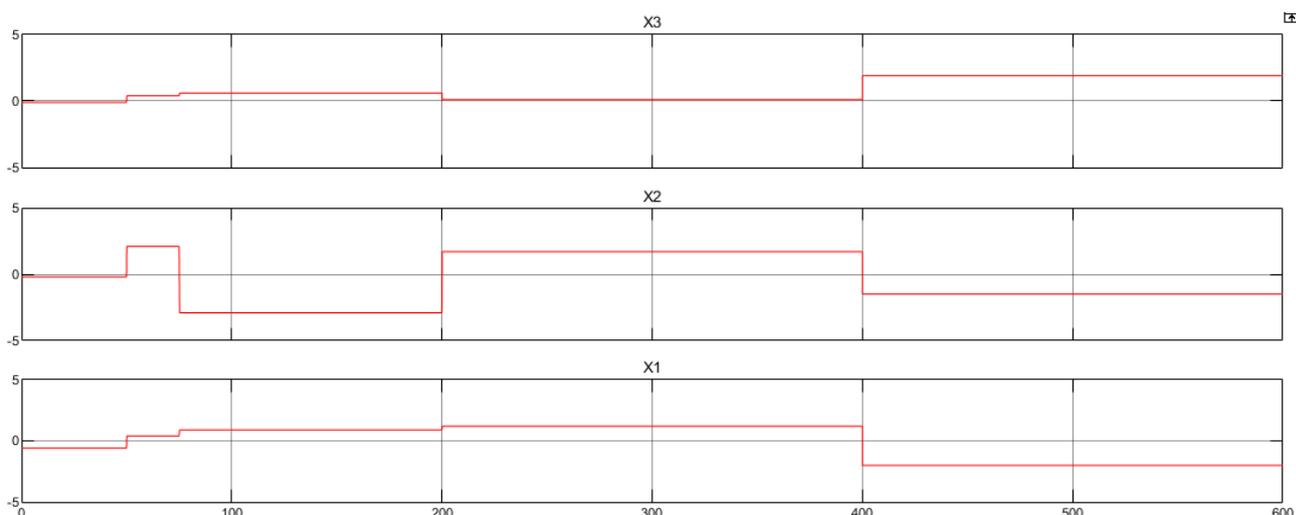


Рисунок 9 – График переключения параметров объекта управления

Согласно рисунку 9, параметры объекта управления в моменты времени: 50 сек., 75 сек., 200 сек. и 400 сек. менялись на другие, важно отметить, что система моментально переключалась между объектами без задержек времени.

При исследовании поведения динамики объекта управления в результате функционирования исследуемой системы, были получены следующие данные, представленные на рисунке 10.

Проанализировав выходные характеристики ОУ и ЭМ, можно сделать вывод, что выходные сигналы совпадают. Следовательно, адаптивный регулятор, синтезированный на этапе разработки системы управления, формирует желаемый управляющий сигнал, приводящий объект управления к эталонной модели, при этом введенная в алгоритмы настройки зона нечувствительности сглаживает резкие скачки параметров объекта управления, что полностью удовлетворяет поставленной цели синтеза и моделирования. Как можно заметить, уровень совпадения динамик ОУ и ЭМ очень высок, что говорит о хорошей работоспособности контура адаптации, поскольку различие выхода ОУ можно заметить только в 4 знаке, как показано на рисунке 11.

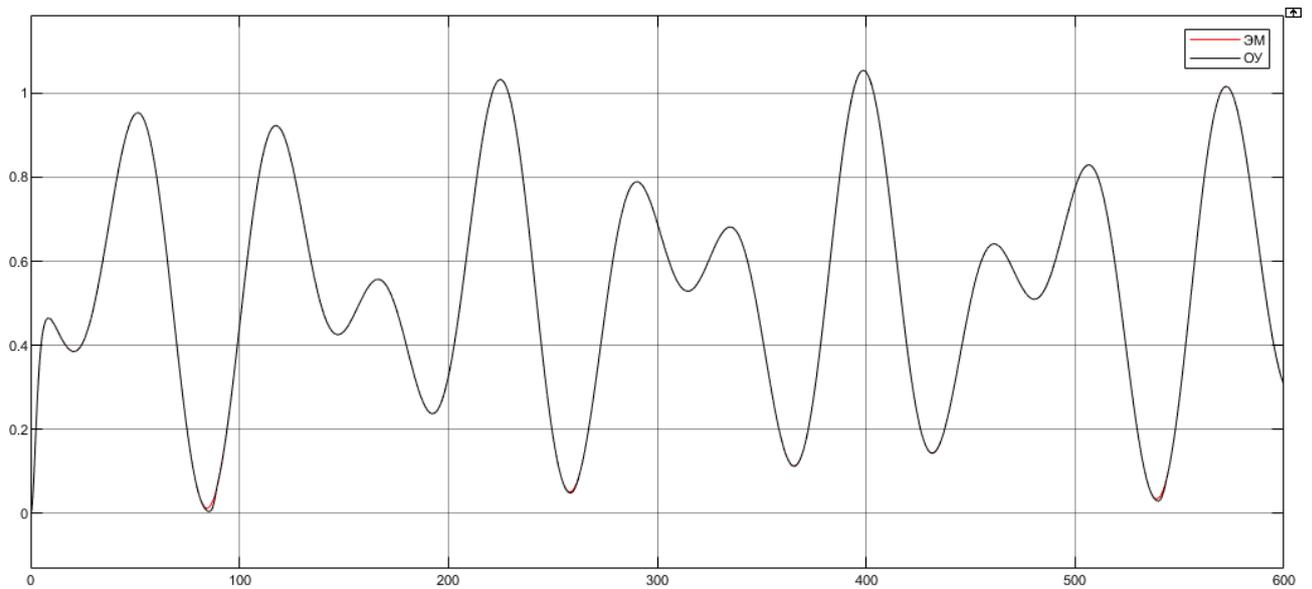


Рисунок 10 – График динамики выхода объекта управления и эталонной модели

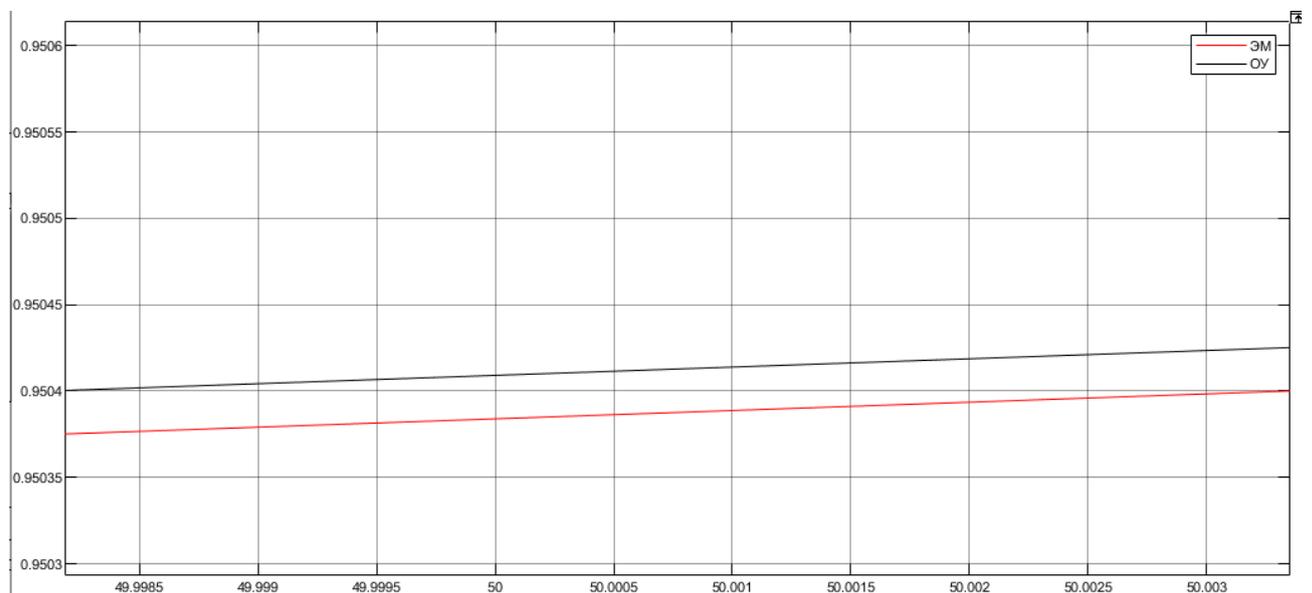


Рисунок 11 – График динамики выхода объекта управления и эталонной модели при увеличении

При измерении уровня ошибки в процессе функционирования системы управления, был получен график рассогласования динамики выхода объекта управления и эталонной модели, показанный на рисунке 12. При этом, анализируя график, можно увидеть, что ошибка уменьшается с течением времени, при вхождении ОУ в установившийся режим. Принимая во внимание особенности ОУ, меняющего свои параметры в один из моментов времени  $q$ , можно заметить резкий рост ошибки, в пике достигающий 0.015, но быстро затухающий в

следствии работы адаптивного регулятора и модифицированных алгоритмов адаптации, в частности введении зоны нечувствительности.

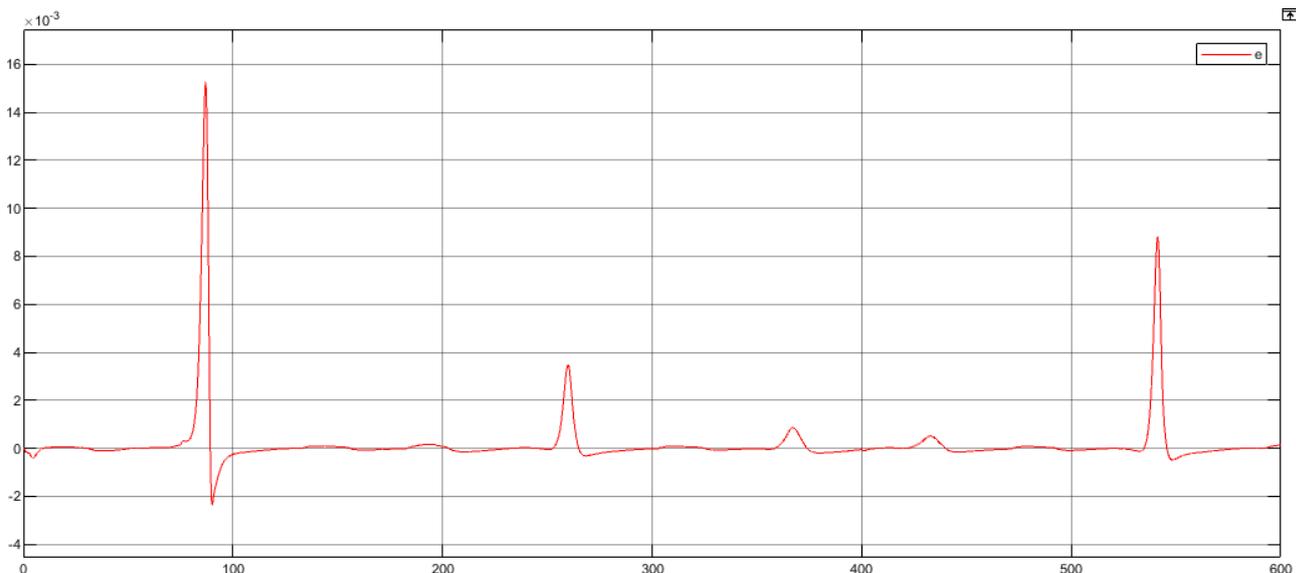


Рисунок 12 – Ошибка рассогласования выхода ОУ и ЭМ

Для анализа формируемого управляющего воздействия адаптивным регулятором был получен следующий график, показанный на рисунке 13. Согласно рисунку, основное воздействие на объект управления происходит в момент начала функционирования системы, это обосновано различием между динамикой ОУ и ЭМ, а также резкие скачки амплитуды сигнала заметны в моменты переключения параметров ОУ.

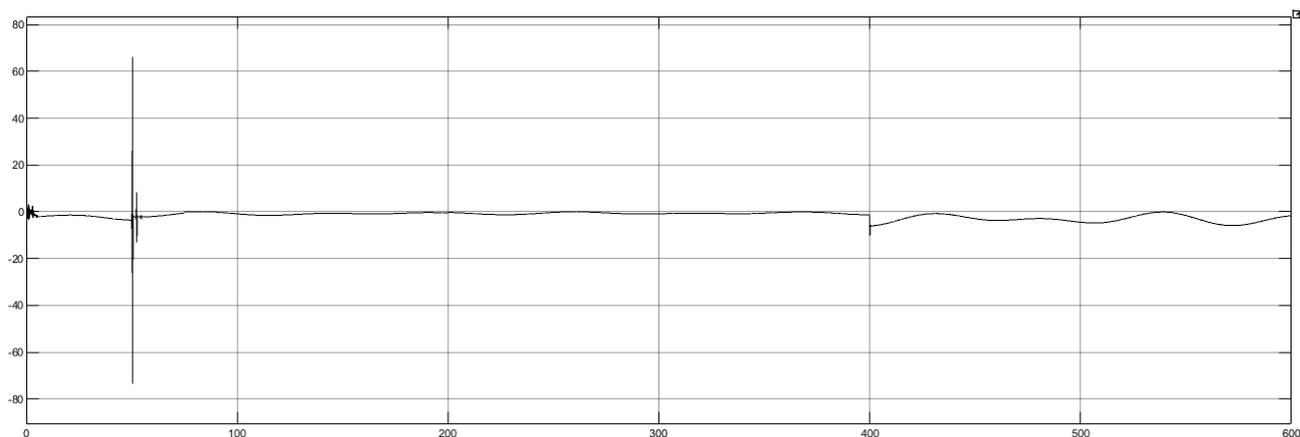


Рисунок 13 – Управляющее воздействие

Анализ работоспособности адаптивного регулятора заключается в исследовании изменения коэффициентов регулирования в процессе имитационно моделирования. Согласно поставленным целям управления, требуется чтобы

каждый из коэффициентов в установившемся режиме стремился к некоторой константе.

На рисунках 14 и 15 представлены графики изменения коэффициентов регулирования в процессе функционирования системы. Согласно рисункам, цели управления были достигнуты, поскольку каждый из коэффициентов стремится к некоторой константе.

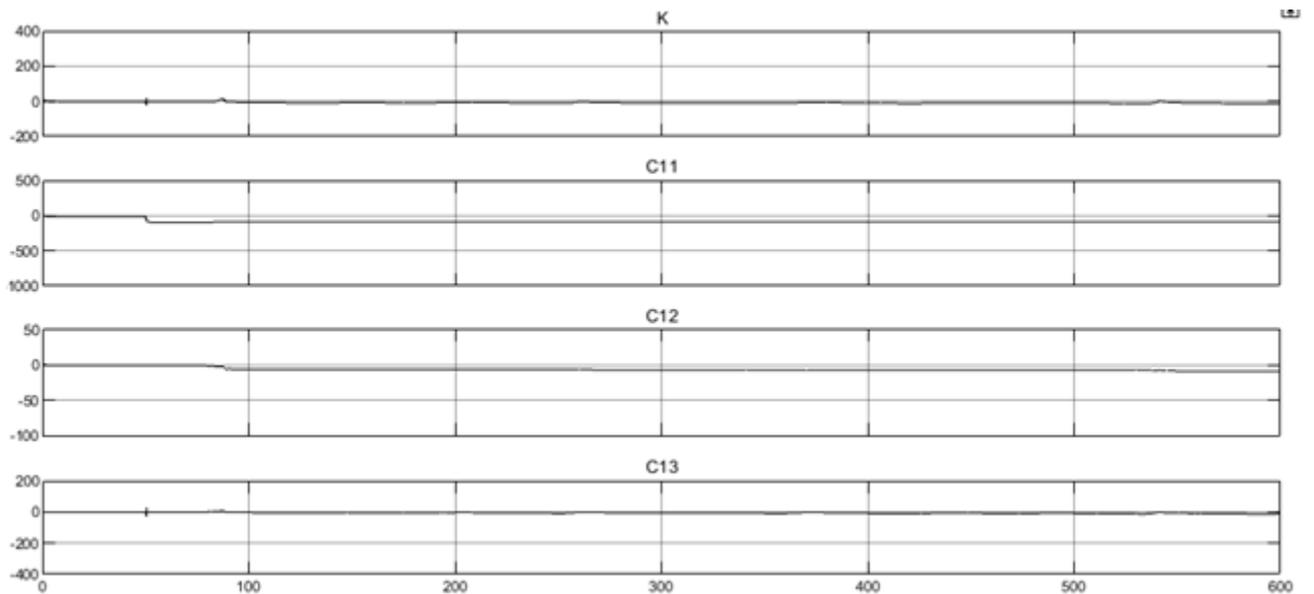


Рисунок 14 – Коэффициенты регулятора

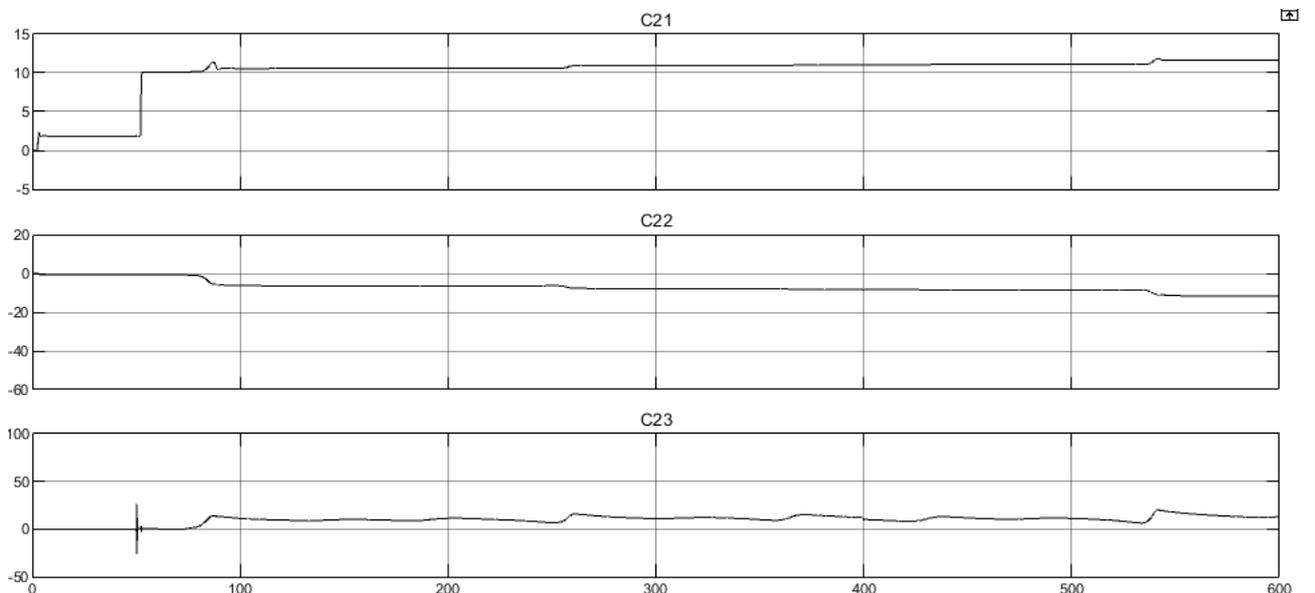


Рисунок 15 – Коэффициенты регулятора по запаздыванию

В результате исследования полученных имитационных характеристик можно сделать вывод, что критерий гиперустойчивости Попова позволяет синтезировать качественные алгоритмы настройки адаптивного регулятора, и по-

лучить работоспособную систему адаптивного управления. Точность же работы системы зависит от настройки коэффициентов АР, получаемые в результате имитационного моделирования. Данный метод не является универсальным, поскольку числовые значения параметров адаптивного регулятора для первичной настройки определяются вручную, так как не существует универсального метода вычисления. Кроме того, границы зоны нечувствительности так же подбирается с учетом динамики объекта управления и задающего воздействия.

Таким образом, в результате проведения анализа синтезированной адаптивной системы управления и исследовании характеристик системы, были подтверждены предположения о работоспособности системы при модификации объекта управления с множеством состояний функционирования при помощи динамического корректора. При этом контур адаптации обеспечил выполнение поставленных целей и условий управления.

## 4 РАЗРАБОТКА ПРИКЛАДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

### 4.1 Обоснование разработки прикладного программного обеспечения

Для работы с системой управления требуется глубокое изучение предметной области, поскольку для управления системой пользователь напрямую взаимодействует с Simulink-схемой. Для управления системой требуется точная настройка таких блоков системы как:

- адаптивный регулятор;
- эталонная модель системы управления;
- динамический корректор;
- объект управления системой;
- генератор задающего воздействия;
- генератор возмущения системы;
- запаздывание в контуре управления.

Кроме того, одним из основных недостатков работы с компонентами MATLAB является локализация программных средств на рабочей станции пользователя – иными словами, для проведения имитационного моделирования требуется установка на рабочей станции пользователя мощного прикладного программного обеспечения пакета MATLAB. Установка требуемого программного обеспечения подразумевает наличие достаточных вычислительных мощностей рабочих станций.

Таким образом, ввиду имеющихся недостатков использования средств моделирования MATLAB, а именно:

- наличие рабочих станций с достаточной вычислительной мощностью, требуемой для корректного функционирования пакета прикладных программ MATLAB;
- наличие обученного персонала, способного работать со сложным интерфейсом MATLAB и Simulink;
- отсутствие предварительного анализа результатов моделирования Simulink системы;

– большие трудозатраты для предварительного расчета параметров системы для моделирования, а также настройка всех блоков Simulink-схемы;

– отсутствие средств сохранения результатов эксперимента в формате, воспринимаемом другими программными средствами, поскольку \*.mat файлы являются зашифрованными текстовыми документами, используемыми только в среде MATLAB,

было принято решение разработать программный продукт решающий следующие задачи:

- клиент-серверная архитектура вычисления;
- автоматизация расчета параметров системы управления;
- автоматизация настройки блоков Simulink-схемы;
- предварительный анализ работоспособности контура адаптации;
- сохранение результатов моделирование в требуемом формате;
- информативное представление результатов моделирования;
- простой и информативный интерфейс, направленный на решение прикладной задачи.

## **4.2 Выбор модели жизненного цикла программного обеспечения**

Жизненный цикл – это модель создания и использования программной системы. Он отражает различные состояния программной системы, начиная с момента возникновения необходимости в этой программной системе и принятия решения о ее создании и заканчивая полным изъятием программной системы из эксплуатации. Международный стандарт ГОСТ 19.201–78 регламентирует структуру жизненного цикла, содержащую процессы, действия и задачи, которые должны быть выполнены во время создания программного обеспечения. Согласно данному стандарту жизненный цикл программного обеспечения базируется на трех группах процессов:

1 основные процессы жизненного цикла, то есть приобретение, поставка, разработка, эксплуатация и сопровождение;

2 вспомогательные процессы, обеспечивающие выполнение основных процессов, то есть документирование, верификация. Аттестация, оценка качества и другие;

3 организационные процессы, то есть управление проектами, создание инфраструктуры проекта и обучение.

Разработка включает в себя все работы по созданию программного обеспечения в соответствии с заданными требованиями. Сюда включается оформление проектной эксплуатационной документации, подготовка материалов, необходимых для проверки работоспособности и качества программных продуктов.

Основные этапы процесса разработки:

- 1 анализ требований заказчика;
- 2 проектирование;
- 3 реализация.

Процесс эксплуатации включает в себя работы по внедрению программного обеспечения в эксплуатацию, в том числе конфигурирование рабочих мест, обучение персонала, локализация проблем эксплуатации и устранение причин их возникновения, модификация программного обеспечения в рамках установленного регламента и подготовка предложения по модернизации системы. Каждый процесс характеризуется определенными задачами и методами их решения, а также исходными данными и результатами. Жизненный цикл программного обеспечения носит, как правило, итерационный характер, то есть реализуются этапы, начиная с самых ранних, которые циклически повторяются в соответствии с изменением требований внешних условий и введением ограничений.

Существует несколько моделей жизненного цикла, которые определяют порядок исполнения этапов разработки и критерии перехода от этапа к этапу. К настоящему времени наибольшее распространение получили две модели жизненного цикла: каскадная и спиральная.

В существующих ранее однородных информационных системах каждое приложение представляло собой единое целое. Для разработки таких приложений применялась каскадная модель жизненного цикла, которую также называют классической или водопадной.

При использовании каскадной модели разработка рассматривалась как последовательность этапов, причем переход на следующий более низкий этап происходит только после того, как полностью завершены все работы на текущем этапе. Подразумевается, что в каскадной модели разработка начинается на системном уровне и происходит через анализ, проектирование, кодирование, тестирование и сопровождение.

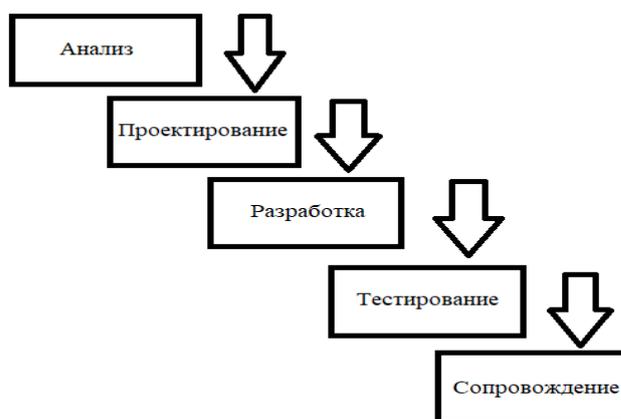


Рисунок 16 – Основные этапы разработки каскадной модели

При анализе задается роль каждому элементу в компьютерной системе и взаимодействия элементов друг с другом. Необходимость системного анализа явно проявляется, когда формируется интерфейс программного обеспечения с другими элементами, то есть с аппаратурой или базами данных. На этом этапе начинается решение задач планирования проекта. В ходе планирования определяется объем проектных работ и их риск, необходимые трудозатраты, формируются рабочие задачи и план-график работ.

Проектирование состоит в создании:

- архитектуры программного обеспечения;
- модульной структуры программного обеспечения;

- алгоритмической структуры программного обеспечения;
- структуры данных;
- входного\выходного интерфейса.

При решении задач проектирования основное внимание уделяется качеству будущего программного продукта.

Разработка или кодирование состоит в переводе результатов проектирования в программный код.

Тестирование – выполнение программы на выявление дефектов в функциях, логике и форме реализации программного продукта.

Сопровождение заключается в внесении изменений в эксплуатируемое программное обеспечение с целью исправления ошибок, адаптации к изменениям внешней для программного обеспечения среды и усовершенствовании функций в соответствии с требованиями заказчика.

К достоинствам применения каскадной модели можно отнести:

- наличие плана и временного графика по всем этапам проекта, что упорядочивает ход разработки программного обеспечения;
- на выходе каждого этапа формируется законченный набор проектной документации, проверенный на полноту и согласованность;
- выполняемые в логической последовательности этапы разработки позволяют планировать сроки завершения всех работ и соответствующие затраты.

Каскадная модель хорошо зарекомендовала себя при построении информационных систем, для которых возможно в самом начале разработки точно сформулировать все требования к системе.

К недостаткам каскадной модели можно отнести:

- реальные проекты часто требуют отклонений от стандартной последовательности шагов;
- каскадная модель основана на точной формулировке исходных требований к программному обеспечению, однако реально в ряде случаев в начале проекта требования заказчика определены только частично;

– результаты реализации проекта доступны заказчику только после завершения всех работ.

Спиральная модель является классическим примером эволюционной стратегии конструирования программного обеспечения, она базируется на лучших свойствах каскадной модели жизненного цикла, к которым добавляется анализ риска.

Спиральная модель включает четыре основных этапа, которые периодически повторяются:

- планирование – определение целей, ограничений и вариантов;
- анализ риска – распознавание риска и анализ вариантов;
- конструирование – это разработка программного продукта следующего уровня;
- оценивание – это оценка заказчика текущих результатов конструирования.

При движении по спирали строятся все более полные версии программного обеспечения при продвижении от центра к периферии. В первом витке спирали определяются начальные цели, варианты и ограничения, распознается и анализируется риск. Если анализ риска показывает неопределенность требований, то используется макетирование, далее заказчик оценивает конструкторскую работу и вносит предложения по модификации. Следующая фаза планирования и анализа риска базируются на предложении заказчика. Если риск слишком велик, проект может быть остановлен. В большинстве случаев движение по спирали продолжается, с каждым шагом продвигая разработчиков к более общей модели системы, схематичное изображение данного жизненного цикла представлено в приложении В.

Достоинства спиральной модели:

- заказчик получает доступ к прототипам программного обеспечения в процессе разработки;
- допускает изменение требований при разработке информационной системы;

- обеспечивает большую гибкость в управлении проектом;
- позволяет совершенствовать процесс разработки;
- уменьшает риски заказчика.

К недостаткам данной модели можно отнести:

- увеличивается неопределенность у разработчика в перспективах развития проекта;
- затруднены операции временного и ресурсного планирования всего проекта в целом.

Таким образом, изучив основные модели жизненного цикла, для разработки программного обеспечения в рамках курсового проекта была выбрана каскадная модель, поскольку, опираясь на предметную область, возможно определить все требования, предъявляемые к разрабатываемому продукту, что является основным критерием для выбора. Кроме того, все недостатки каскадной модели не повлияют на качество разработки требуемого программного обеспечения.

### **4.3 Описание функций программного обеспечения**

При разработке прикладного программного обеспечения управления системой была разработана следующая функциональная диаграмма, представленная в приложении Г на рисунке Г1.

Согласно данной диаграмме, входными данными для ПО должны являться:

- данные пользователя;
- параметры системы;
- база данных.

Данные пользователя требуются для авторизации, что позволит обеспечить безопасность использования программного продукта, а также позволит сохранять сессионную информацию каждого пользователя, что делает ПО многопользовательской программой. Параметры системы являются данными, введенными пользователем, они используются для расчета коэффициентов системы управления, которые, в свою очередь, передаются в схему для имитацион-

---

ного моделирования. База данных позволяет загружать и сохранять сессионную информацию, используемую пользователем.

Результатом функционирования ПО является:

- отчеты функционирования системы управления;
- характеристики системы;
- измененная база данных.

Программное обеспечение позволяет формировать различные отчеты, после имитационного моделирования, для дальнейшего анализа функционирования системы управления. Для наглядности происходящих процессов в системе при имитационном моделировании будут строиться графики полученных характеристик системы управления в результате моделирования. При завершении работы с программным модулем, вся полученная информация будет сохраняться в базу данных.

Главными управляющими механизмами для приложения будут ограничения системы управления и техническое задание, на основании которого прикладное программное обеспечение будет производит расчет параметров системы по введенным пользователем данным.

Для функционирования приложения, кроме встроенных возможностей языка программирования MATLAB, будет использована технология «Workspace», позволяющая в реальном времени обеспечить обмен информацией с Simulink-схемой, а также технология «COM», для формирования различных отчетов в прикладных текстовых и графических программах.

При разработке, были выделены следующие функциональные подсистемы, выполняющие основные функции работы приложения для управления системой. Декомпозиция представлена в приложении Г на рисунке Г2.

Первой функциональной подсистемой приложения является подсистема Авторизации пользователя. Авторизация пользователя обеспечит безопасность данных, хранимых модулем, поскольку разграничит доступ пользователей.

Вторая функциональная подсистема приложения – система Задания параметров. Для пользователей, прошедших авторизацию, будет доступна настрой-

ка параметров, что включает в себя ручной ввод коэффициентов, либо загрузка из базы данных ранее сохраненных. При ручном вводе, подсистема будет проверять ввод данных на корректность согласно ограничениям системы управления, в случае неверных данных предусмотрена возможность оповещения пользователя об ошибке.

Третьей функциональной подсистемой является Расчет необходимых параметров адаптивной системы управления. После ввода параметров, система на основе введенных данных рассчитывает динамический корректор, алгоритм расчета представлен в приложении А.

Четвертой функциональной подсистемой является Загрузка параметров в ядро MATLAB. Данный модуль выполняет передачу параметров, введенных пользователем и рассчитанных, ядру MATLAB для выполнения моделирования динамики системы управления.

Пятая функциональная подсистема – Моделирование. Данная подсистема после передачи данных в рабочую область, загружает имитационную модель и автоматически запускает систему управления с введенными ранее параметрами.

Шестой функциональной подсистемой является Выгрузка результатов моделирования из ядра MATLAB. Данная подсистема после моделирования передает все полученные результаты приложению, для отображения характеристик системы.

Заключительной функциональной подсистемой является подсистема Формирования отчетов. Данная подсистема позволяет получить информацию о функционировании системы управления в виде различных отчетов и графиков. Данная подсистема основана на технологии «СОМ», которая позволяет задействовать все имеющиеся текстовые и графические редакторы.

Таким образом, данная структура программного обеспечения для управления системой позволит выполнять все поставленные задачи.

## 4.4 Архитектурный проект программного обеспечения

### 4.4.1 Диаграмма вариантов использования

Диаграмма вариантов использования описывают взаимоотношения и зависимости между группами вариантов использования и действующими лиц, участвующих в процессе.

Для построения диаграммы вариантов использования, разрабатываемого прикладного программного обеспечения, были выделены следующие группы лиц:

- пользователь (оператор);
- администратор системы.

Для выделенных групп лиц была построена следующая диаграмма вариантов использования, представленная в приложении Д на рисунке Д1.

Согласно представленной диаграмме, администратор устанавливает необходимое программное обеспечение на рабочей станции пользователя, а именно:

- MATLAB 2012b или выше (в случае отсутствия сервера, где установлен MATLAB);
- Microsoft .NET Framework 4.6 или выше;
- Необходимые драйвера принтеров и текстовые редакторы, для вывода отчетов.

Так же администратор настраивает подключение приложения к серверу баз данных MySQL и серверу MATLAB. После настройки приложения, администратор добавляет или изменяет данные пользователя и его права. Администратор может дублировать все действия пользователя приложения.

Пользователь, получивший логин и пароль от администратора, имеет право на авторизацию в приложении и последующей работе с системой управления. В ходе работы пользователь может настраивать параметры моделируемой системы управления, производить имитационное моделирование для получения характеристик системы, работать с результатами моделирования.

#### 4.4.2 Диаграмма последовательности

Диаграммы последовательности являются видом диаграмм взаимодействия, которые описывают отношения объектов в различных условиях. Условия взаимодействия задаются сценариями, полученными на этапе разработки диаграмм вариантов использования.

Сценарий взаимодействия пользователя с разрабатываемым прикладным программным обеспечением представлен в приложении Д на рисунке Д2.

Согласно построенной диаграмме пользователь при запуске прикладного программного обеспечения должен пройти аутентификацию и авторизацию. При успешном прохождении процедур авторизации и аутентификации пользователь настраивает параметры системы путем ручного ввода, либо загрузке ранее сохраненных параметров. При некорректных введенных данных приложение сообщит об ошибке.

На следующем шаге, используя предварительные данные введенные пользователем, система выполняет расчет необходимых для моделирования параметров, в случае ошибки, приложение инициализирует исключительную ситуацию и выдаст соответствующее сообщение пользователю.

После успешного расчета всех параметров, пользователь может запустить процедуру имитационного моделирования, приложение по команде свяжется с сервером и передаст все параметры, требуемые для моделирования, ядру MATLAB. В случае возникновения ошибки связи с сервером MATLAB или ошибки передачи параметров серверу, приложение сгенерирует соответствующее сообщение пользователю об ошибке.

Следующим шагом, после передачи параметров ядру MATLAB, приложение посылает команду для получения результатов и ожидает ответа от сервера, который производит необходимые расчеты. При ошибке серверной части, приложение сообщит пользователю об ошибке моделирования.

При получении ответа от сервера о успешные имитационные моделирования системы управления, с введенными и рассчитанными параметрами, приложение запрашивает все имитационные характеристики, полученные ядром

MATLAB. При возникновении ошибки приложение выдаст сообщение об ошибке выгрузки параметров из ядра MATLAB.

Заключительным этапом работы с разработанным прикладным программным обеспечением является визуализация полученных результатов моделирования и формирование пользователем отчетов о произведенном моделировании. При возникновении ошибки на этапе формирования отчетов, приложение сгенерирует исключительную ситуацию и выдаст подробное сообщение пользователю.

Таким образом, была спроектирована диаграмма последовательности действий пользователя (оператора) при работе с разрабатываемым прикладным программным обеспечением.

#### 4.4.3 Диаграмма состояний

Диаграмма состояний представляет из себя ориентированный граф, который определяет множество состояний, в котором может находится приложение. Главное назначение этой диаграммы – описать все возможные последовательности состояний и переходов, которые в совокупности характеризуют поведение элемента модели в течение его жизненного цикла.

Разработанная диаграмма состояний разрабатываемого программного обеспечения представлена в приложении Д на рисунке ДЗ.

Согласно диаграмме ДЗ в приложении Д, разрабатываемое программное обеспечение имеет 7 основных состояний.

«Авторизация» – состояние при котором приложение ожидает ввода логина и пароля пользователя для определения прав доступа к системе. При успешной авторизации пользователя приложение перейдет в состояние «Задание параметров системы управления». Так же пользователь может выйти из приложения на этапе авторизации. В случае возникновения ошибки приложение останется в состоянии «Авторизация».

«Задание параметров системы управления» – состояние при котором приложение ожидает ввода предварительных параметров от пользователя, либо загрузку ранее заданных параметров из базы данных. После ввода предваритель-

ных данных приложение переходит в состояние «Расчет дополнительных параметров». При введении некорректных данных, система остается в состоянии «Задание параметров системы управления» и выдает соответствующее сообщение пользователю. Пользователь может завершить работу приложения находясь на стадии «Задание параметров системы управления».

«Расчет дополнительных параметров» – состояние при котором приложение на основании введенных пользователем или загруженных из базы данных параметров рассчитывает постоянную времени для динамического корректора. При возникшей на этой стадии ошибке, приложение переходит в состояние «Задание параметров системы управления» и выдает соответствующее сообщение пользователю. При успешном расчете приложение переходит в состояние «Передача данных в MATLAB».

«Передача данных в MATLAB» – состояние при котором приложение соединяется с сервером, где установлено пакет прикладных программ MATLAB, после успешного соединения происходит передача данных для моделирования ядру. При возникновении ошибки приложение переходит в состояние «Задание параметров системы управления» и выдает соответствующее сообщение пользователю. При успешной передаче необходимых параметров, приложение инициализирует команду для имитационного моделирования системы управления и переходит в состояние «Ожидание имитационного моделирования на сервере».

«Ожидание имитационного моделирования на сервере» – состояние при котором приложение ожидает завершения процесса имитационного моделирования на сервере, при этом реагируя на сообщения от сервера, при зависании серверной части, возникновении ошибки на стадии моделирования, приложение генерирует исключительную ситуацию, принудительно завершает работу сервера и переходит в состояние «Задание параметров системы управления», при этом пользователь получит соответствующее сообщение об ошибке. При успешном имитационном моделировании приложение перейдет в состояние

«Визуализация результатов моделирования, формирование требуемых отчетов».

«Визуализация результатов моделирования, формирования требуемых отчетов» – состояние при котором приложение визуализирует в виде графиков имитационные характеристики, полученные в результате расчетов сервера. При этом пользователь может сохранить необходимые данные в удобной для дальнейшей обработки форме. При возникновении ошибки, на стадии визуализации характеристик, приложение перейдет в состояние «Задание параметров системы управления» и сгенерирует исключительную ситуацию, пользователь получит соответствующее сообщение об ошибке.

Далее пользователь может самостоятельно перевести приложение в состояние «Задание параметров системы» и повторить все этапы моделирования.

#### 4.4.4 Диаграмма активности

Диаграмма активности – диаграмма, показывающая спецификацию исполняемого поведения в виде координированного последовательного и параллельного выполнения подчиненных элементов системы.

Для разрабатываемого прикладного программного приложения была построена диаграмма активности, представленная в приложении Д на рисунке Д4.

Согласно диаграмме, были выделены 3 объекта, непосредственно участвующие при работе с программным средством, а именно: пользователь, клиентская часть приложения и серверная часть приложения.

Как видно из диаграммы, пользователь, после прохождения авторизации и аутентификации инициализирует ввод (корректировку) параметров системы управления, после чего управление на себя берет клиентская часть приложения. Клиентская часть проверяет корректность ввода данных пользователем, и при успешной проверке автоматически рассчитывает параметры динамического корректора, согласно алгоритму, представленному в приложении А. После успешного расчета все данные передаются серверной части приложения, где и происходит имитационное моделирование адаптивной системы управления с заданными пользователем параметрами. После успешного моделирования сер-

верная часть приложения передает все полученные характеристики системы на клиентскую часть, на которой происходит визуализация требуемых данных, формирование отчетов и сохранение результатов в базу данных. Серверной частью приложения выступает рабочая станция с сервером баз данных и установленным пакетом прикладных программ MATLAB, который и выполняет все требуемые векторно-матричные вычисления.

#### 4.4.5 Диаграмма классов

Для демонстрации общей структуры иерархии классов системы, атрибутов и методов, а также взаимодействие между ними была построена диаграмма классов, представленная в приложении Д на рисунке Д5.

Согласно диаграмме классов, для обеспечения приложения всем требуемым функционалом, а также для обеспечения основного принципа программирования – объектно-ориентированность, были спроектированы следующие классы и интерфейсы:

- класс «DBUtils»;
- класс «BDMySQLUtils»;
- класс формы «MainForm»;
- класс формы «Autorisation»;
- класс «User»;
- интерфейс «IOObjectSimulink»;
- класс «DK»;
- класс «RRParam»;
- класс «RG»;
- класс «EM»;
- класс «OU».

Классы «DBUtils» и «BDMySQLUtils» отвечают за взаимодействия приложения с базой данных.

Класс «MainForm» отвечает за функционирование основной формы приложения, содержит методы, отвечающие за получение данных о пользователе, параметрах регулятора, параметрах эталонной модели и параметрах объекта

управления. Все параметры задаются пользователем вручную или загружаются из базы данных.

Интерфейс «IOObjectSimulink» описывает поведение всех объектов системы управления.

Классы «DK» и «RRParam» отвечают за формирование параметров динамического корректора, «RRParam» содержит методы для расчета постоянной времени согласно алгоритму, описанному в приложении А.

Класс «RG» отвечает за формирование коэффициентов адаптивного регулятора.

Класс «EM» формирует полное описание эталонной модели.

Класс «OU» отвечает за формирование параметров объекта управления.

## **4.5 Описание реализации прикладного программного обеспечения**

### **4.5.1 Описание среды разработки приложения**

Для разработки требуемого приложения была выбрана среда Microsoft Visual Studio 2017 Community. Данная среда является свободно распространяемой, интегрированной средой разработки программного обеспечения имеющая ряд инструментальных средств, таких как:

- редактор исходного кода, включающий множество дополнительных функций, как авто дополнение IntelliSense, рефакторинг кода, автоматический отладчик;

- редактор форм, предназначенный для упрощенного конструирования графических интерфейсов;

- веб-редактор, позволяющий удаленную разработку приложений;

- дизайнер классов;

- дизайнер схемы базы данных.

Среда разработки Visual Studio представляет собой полный набор инструментов для создания как настольных приложений, так и корпоративных веб-приложений для совместной работы групп. Используя эффективные инструменты разработки Visual Studio, основанные на использовании компонентов, и другие технологии, можно не только создавать эффективно работающие

настольные приложения, но и упрощать совместное проектирование, разработку и развертывание корпоративных решений [15-17].

Для конкретной реализации была выбрана связка двух технологий:

– WPF – Windows Presentation Foundation (WPF) представляет собой обширный API-интерфейс для создания настольных графических программ имеющих насыщенный дизайн и интерактивность. В отличие от устаревшей технологии Windows Forms, WPF включает новую модель построения пользовательских приложений (в основе WPF лежит мощная инфраструктура, основанная на DirectX). Это означает возможность применения развитых графических эффектов без потери производительности, что является основным преимуществом над Windows Forms. Данная технология позволяет использовать такие расширенные средства, как поддержка видеофайлов и трехмерное содержимое. Используя эти средства в связке с развитым графическим редактором, возможно создавать современные пользовательские интерфейсы и визуальные эффекты, которые были недоступны при использовании устаревшей технологии Windows Forms.

– COM (Component Object Model) – модель компонентного объекта, позволяющая выполнять расчеты на стороннем приложении MATLAB. Данная технология позволяет применять архитектуру клиент-сервер, что обеспечивает функциональность и масштабируемость разработанного приложения.

#### 4.5.2 Реализация взаимодействия с ядром MATLAB

Используемая технология COM позволяет построить многоуровневую архитектуру клиент-сервер. Многоуровневая архитектура – разновидность архитектуры клиент-сервер, в которой функции обработки данных вынесена на один или несколько отдельных серверов. Это позволяет разделить функции хранения, обработки и представления данных для более эффективного использования возможностей серверов и клиентов.

Приложение, построенное на стандарте COM, представляет из себя набор взаимодействующих между собой COM-компонентов. Каждый компонент имеет уникальный идентификатор (GUID) и может использоваться одновременно

многими программами. Таким образом, на рабочей станции, являющейся сервером, установленный пакет прикладных программ MATLAB имеет уникальный идентификатор GUID, используемыми несколькими клиентскими приложениями, обращающимися к серверу.

Основываясь на используемой технологии COM была реализована трехуровневая архитектура (трехзвенная архитектура), предполагающая наличие следующих компонентов приложения: клиентское приложение, сервер приложений и сервер баз данных. Схематично такую архитектуру можно представить как показано на рисунке 17.

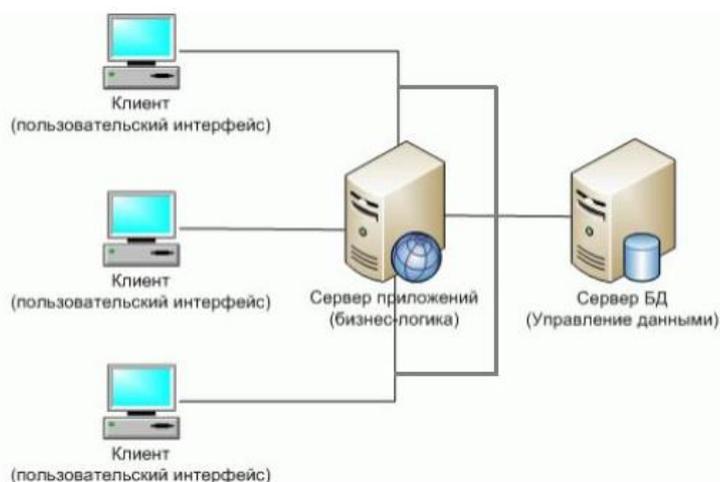


Рисунок 17 – Трехуровневая архитектура

Клиентское приложение – это интерфейсный компонент, который представляет первый уровень, собственно приложение конечного пользователя. Для разработанного программного обеспечения, первый уровень имеет возможность обращаться к БД не только через сервер приложений, но и напрямую, что позволяет снижать нагрузку на сервер приложений при большом количестве клиентов. Так же клиент не нагружен основной бизнес-логикой (по требованиям масштабируемости), все расчеты и моделирование системы происходит на серверной части, клиентское приложение обращается к ядру MATLAB по уникальному идентификатору GUID и производит требуемые расчеты. Так же клиент не хранит состояние приложения по требованиям надежности. Клиент содержит простую бизнес логику для авторизации пользователя, контроль введенных данных пользователем, отображения результатов моделирования.

Сервер приложений располагается на втором уровне. На втором уровне сосредоточена большая часть бизнес-логики. Вне его остаются фрагменты, экспортируемые на клиентские приложения, а также погруженные в третий уровень хранимые процедуры и триггеры. Для разрабатываемого приложения вторым уровнем является рабочая станция установленным пакетом прикладных программ MATLAB.

Сервер баз данных является третьим уровнем архитектуры, он обеспечивает хранение данных. Сервер баз данных возможно разместить на отдельной рабочей станции, для увеличения безопасности системы в целом.

Плюсами данной архитектуры является:

- клиентское ПО не нуждается в администрировании;
- масштабируемость;
- конфигурируемость;
- высокая безопасность;
- высокая надежность;
- низкие требования к производительности клиентов.

#### 4.5.3 Реализация алгоритма расчета имитационных характеристик системы

Одним из основных критериев разработки прикладного программного обеспечения в рамках магистерской диссертации является возможность удаленного использования вычислительных мощностей пакета прикладных программ MATLAB для получения имитационных характеристик синтезируемой системы. Согласно поставленным целям была выбрана и применена технология COM, согласно следующему алгоритму, представленному в приложении Е на рисунке Е1.

Согласно разработанному алгоритму, после инициализации пользователем процедуры расчета имитационных характеристик, приложение проверяет доступность сервера, в случае успешной проверки, создается COM компонент по уникальному идентификатору GUID, которому передаются все параметры, требуемые для имитационного моделирования системы управления. После пе-

редачи всех параметров инициализируется процесс моделирования, после чего клиентское приложения выгружает все требуемые характеристики системы управления и визуализирует их на экране приложения.

В случае недоступности сервера, приложение попытается найти установленный на локальной рабочей станции пользователя пакет прикладных программ и, при наличии его повторит ранее описанный алгоритм действий, в случае недоступности MATLAB, приложение выдаст сообщение об ошибке, поскольку выполнить расчеты имитационных характеристик будет невозможно.

#### 4.5.4 Реализация взаимодействия приложения с базой данных

Согласно трехуровневой архитектуре клиент-серверного приложения, сервер баз данных является третьим и заключительным уровнем. Для обеспечения надежности и производительности системы, для хранения данных используется отдельная рабочая станция, на которой установлена система управления реляционными базами данных, разработанная корпорацией Microsoft – SQL Server Express.

Microsoft SQL Server Express является бесплатно распространяемой версией SQL Server, развитием системы MSDE. Данная версия имеет некоторые технические ограничения. Такие ограничения делают ее непригодной для развертывания больших баз данных, но она вполне годится для ведения программных комплексов в масштабах небольшой компании. Содержит полноценную поддержку новых типов данных, в том числе XML-спецификации.

Microsoft SQL Server в качестве языка запросов использует версию SQL, получившую название Transact-SQL, являющуюся реализацией SQL-92 с множественными расширениями. Transact-SQL позволяет использовать дополнительный синтаксис для хранимых процедур и обеспечивает поддержку транзакций.

Microsoft SQL Server также поддерживает Open Database Connectivity (ODBC) – интерфейс взаимодействия приложения с СУБД. Так же одним из важных критериев является использование полного набора библиотек, доступных для .NET Framework, включая Common Type System (система обращения с

типами данных в Microsoft .NET Framework). Однако, в отличие от других процессов, .NET Framework выделяет дополнительную память и выстраивает средства управления SQL Server вместо того, чтобы использовать встроенные средства Windows. Это повышает производительность в сравнении с общими алгоритмами Windows, так как алгоритмы распределения ресурсов специально настроены для использования в структурах SQL Server.

## 4.6 Руководство пользователя

### 4.6.1 Авторизация

Для работы с разработанным прикладным программным обеспечением требуется регистрация учетной записи администратором системы. При первом запуске приложение, окно авторизации запросит логин и пароль пользователя. На рисунке 18 показан внешний вид окна авторизации.

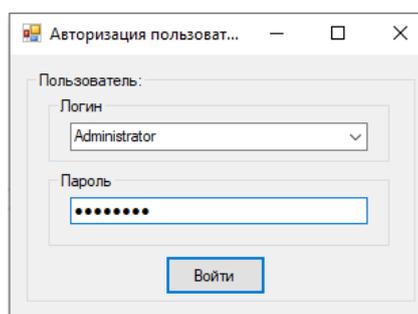


Рисунок 18 – Окно авторизации пользователя

Пользователь может выбрать из выпадающего списка свой логин для определения прав доступа к системе и ввести соответствующий пароль. Символы пароля скрыты для предотвращения захвата экрана сторонними программами. В случае некорректного ввода пароля, пользователь получит соответствующее сообщение об ошибке.

### 4.6.2 Рабочая область

После успешной авторизации пользователя приложение запускает главную форму, являющуюся основной рабочей областью программы, окно рассматриваемой формы представлено на рисунке 19.

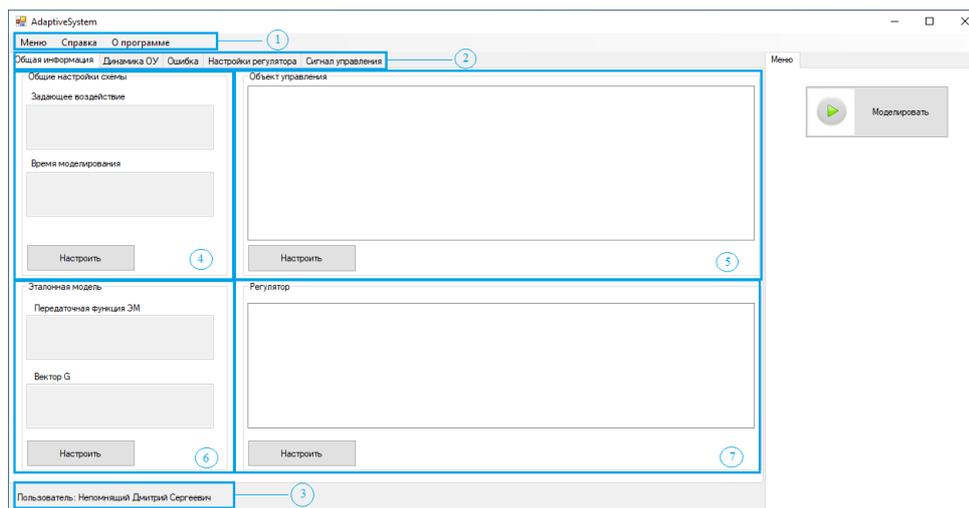


Рисунок 19 – Рабочая область приложения

Согласно изображению, в области под цифрой 1 находятся основные управляющие кнопки приложения: «Меню», «Справка» и «О программе».

Пункт «Меню» позволяет открыть модуль работы с пользователями, а также вывести на печать, всю полученную в процессе моделирования информацию. Модуль работы с пользователями доступен только при наличии прав администратора, внешний вид окна работы с пользователями представлен на рисунке 20.

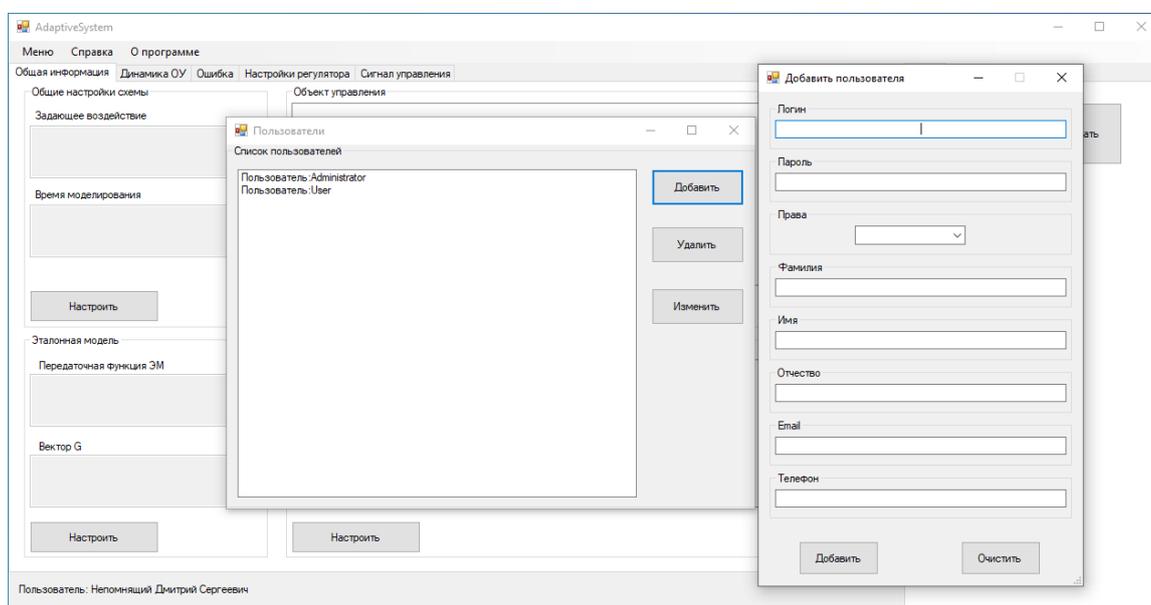


Рисунок 20 – Модуль управления пользователями

Как видно из рисунка 20, модуль управления пользователями содержит список существующих пользователей, зарегистрированных в системе, а также три управляющие кнопки «Добавить», «Удалить» и «Изменить».

При добавлении и изменении пользователя, открывается окно регистрации для ввода данных о пользователе:

- логин;
- пароль;
- права доступа к приложению;
- фамилия;
- имя;
- отчество;
- электронная почта пользователя;
- телефон.

После успешной регистрации, новый пользователь будет добавлен в систему и отражен в списке пользователей.

Кнопки «Удалить» и «Изменить» используются для удаления пользователей из системы и изменения регистрационных данных.

При нажатии на кнопку «Справка» приложение откроет файл справочной информации. Справка содержит основные сведения о функционале приложения в удобной для пользователя форме, так же справка содержит подробное математическое описание синтезируемой адаптивной системы управления, для понимания всех процессов, протекающий при моделировании. Пример окна справки представлен на рисунке 21.

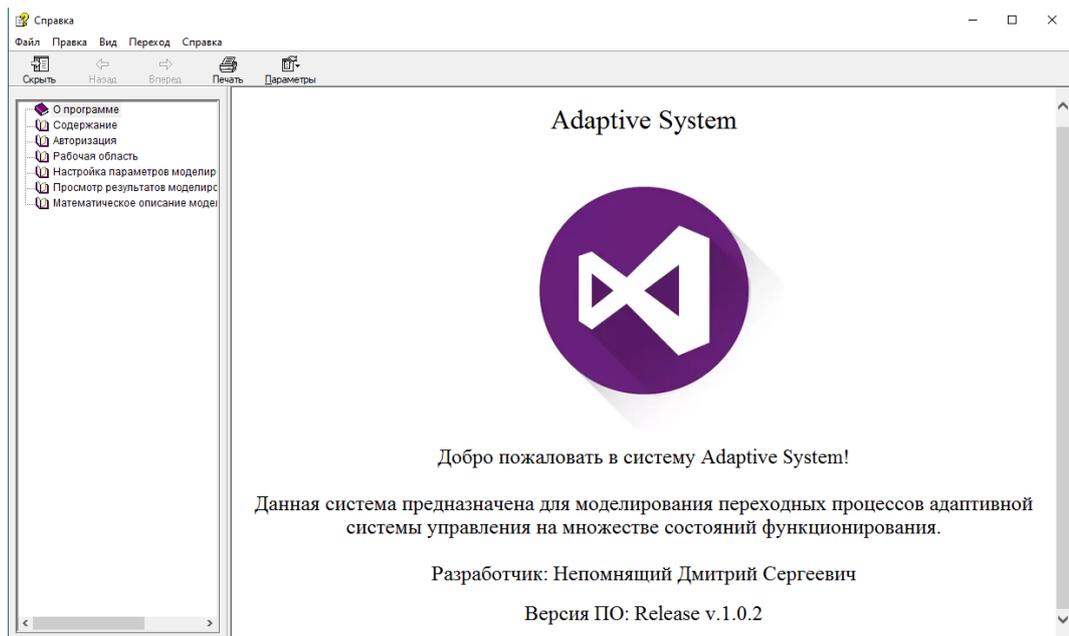


Рисунок 21 – Внешний вид справочной системы

При нажатии на кнопку «О программе» пользователь получит информации о разработчиках и версии текущего программного обеспечения.

На рисунке 19 под цифрой 2 отмечена область переключения между окнами отображения информации.

«Общая информация» – окно где пользователю доступно отображение и настройка всех параметров системы управления, требуемых для имитационного моделирования.

Окна «Динамика ОУ», «Ошибка», «Настройки регулятора» и «Сигнал управления» станут доступны пользователю после завершения процесса моделирования.

Под цифрой 3 на рисунке 19 отмечена область, отображающая данные о пользователе приложения.

Для запуска имитационного моделирования, требуется настроить все параметры в блоках и нажать на кнопку «Моделировать».

#### 4.6.3 Настройка параметров моделирования

Для настройки параметров моделирования, пользователю требуется ввести данные в четырех блоках:

- блок «Общие настройки схемы»;
- блок «Объект управления»;

- блок «Эталонная модель»;
- блок «Регулятор».

Блок «Общие настройки схемы» содержит математическое описание задающего воздействия и параметр – время моделирования. Для задания текущих данных требуется нажать на кнопку «Настроить» текущего блока. На рисунке 22 представлено окно ввода задающего воздействия и времени моделирования.

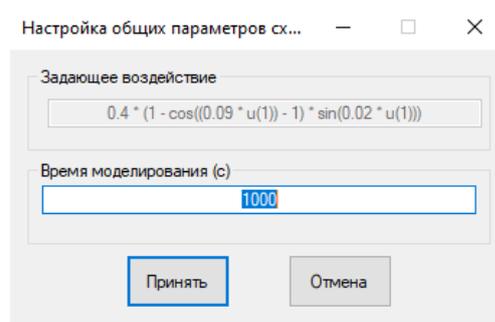


Рисунок 22 – Окно ввода общих параметров схемы

Блок «Объект управления» содержит полное математическое описание объекта управления. После нажатия на кнопку «Настроить» пользователю потребуется задать:

- количество переключений параметров объекта управления ( $q$ );
- максимальную степень передаточной функции объекта управления;
- задать матрицы состояния объекта управления для каждого из состояний ( $q$  раз);
- задать матрицу запаздывания;
- задать временное запаздывание переменных состояния объекта управления.

Блок «Эталонная модель» содержит полное математическое описание эталонной модели, кроме того, в данном блоке задается вектор  $\mathbf{g}$ . Для ввода математического описания эталонной модели, требуется нажать на кнопку «Настроить».

Блок «Регулятор» содержит описание всех коэффициентов адаптивного регулятора. Для ввода требуемых коэффициентов требуется нажать на кнопку «Настроить».

После ввода данных в каждом из блоков, в окне «Общая информация» будет отображаться заданная пользователем информация в соответствующем блоке. Пример окна с введенными данными пользователем представлено на рисунке 23.

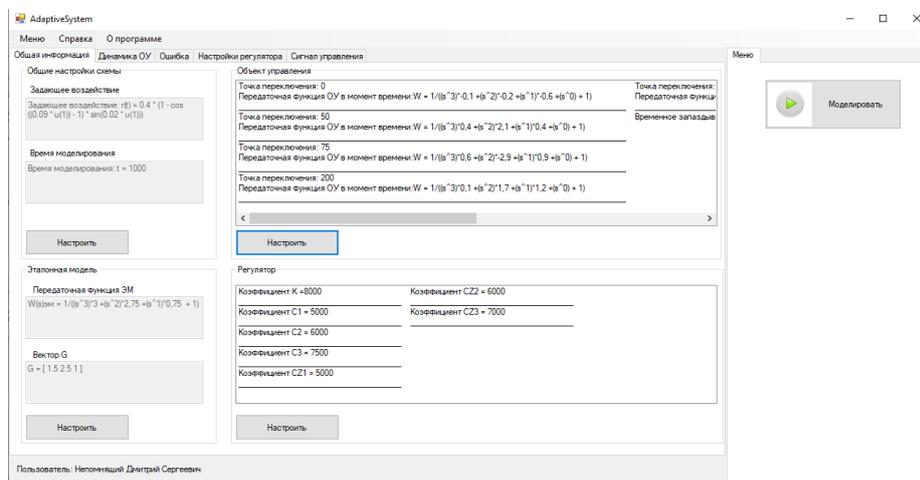


Рисунок 23 – Внешний вид окна с настроенными параметрами в блоках

После настройки всех блоков, пользователю станет доступна кнопка «Моделировать», инициализирующая процесс имитационного моделирования. При инициализации процедуры моделирования, согласно алгоритму в приложении А рассчитываются параметры динамического корректора и только после этого передаются на серверную часть

#### 4.6.4 Просмотр результатов моделирования

После успешного моделирования, пользователю станут доступны окна просмотра полученных имитационных характеристик заданной системы управления.

«Динамика ОУ» – окно, содержащее полученные имитационные характеристики в ходе моделирования. Данное окно подразумевает изучение динамики переходных процессов, протекающих в моделируемой системе управления. Для пользователя доступен широкий инструментарий для работы с полученным графиком:

- масштабирование;
- выделение требуемых зон для детального изучения;
- возможность сохранить график в требуемом формате;

- распечатать требуемый график или область;
- настроить отображение точек на графике.

Пример окна «Динамика ОУ» представлен на рисунке 24.

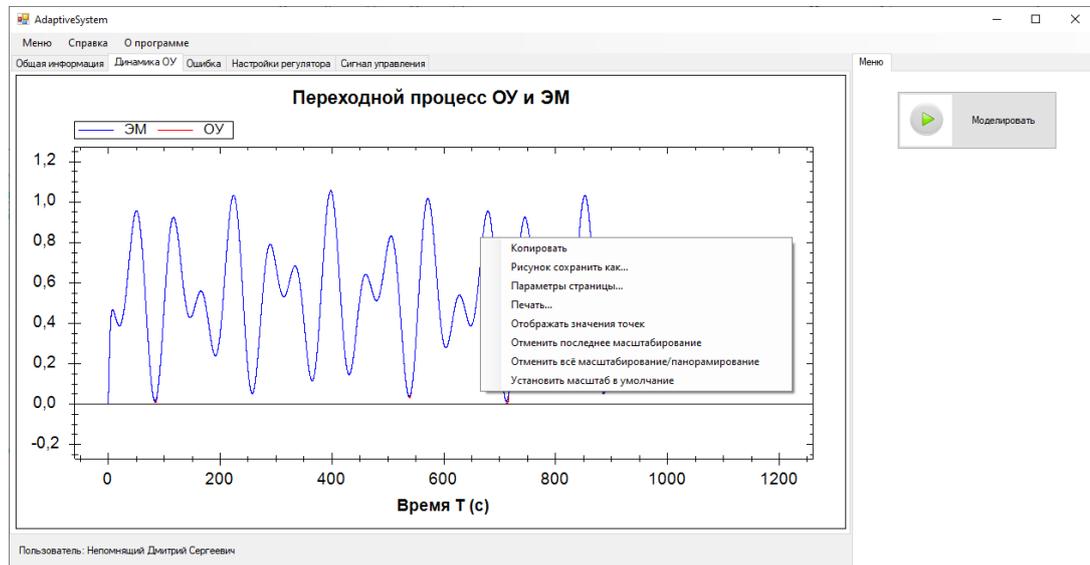


Рисунок 24 – Внешний вид окна «Динамика ОУ»

«Ошибка» – окно, содержащее информацию об ошибке рассогласования между динамикой объекта управления и эталонной модели в процессе имитационного моделирования. Для исследования текущей характеристики системы доступен аналогичный инструментарий, описанный выше. Пример внешнего вида окна «Ошибка» представлен на рисунке 25.

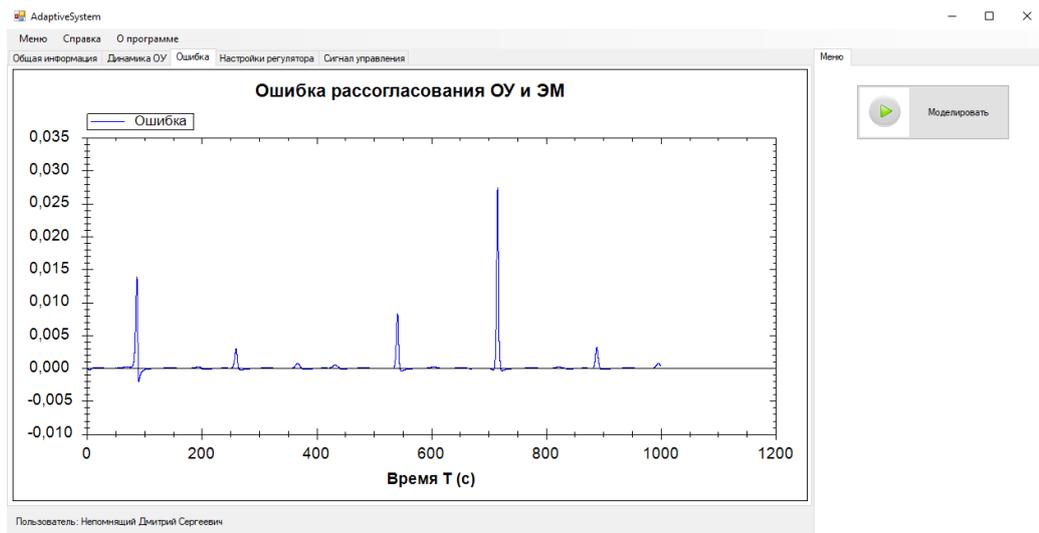


Рисунок 25 – Внешний вид окна «Ошибка»

«Настройки регулятора» – окно, отображающее динамику каждого из коэффициентов адаптивного регулятора, для удобства коэффициенты разбиты на две подгруппы:

- коэффициенты по переменным состояниям;
- коэффициенты по запаздыванию.

Пример визуализации настроек регулятора представлен на рисунке 26.

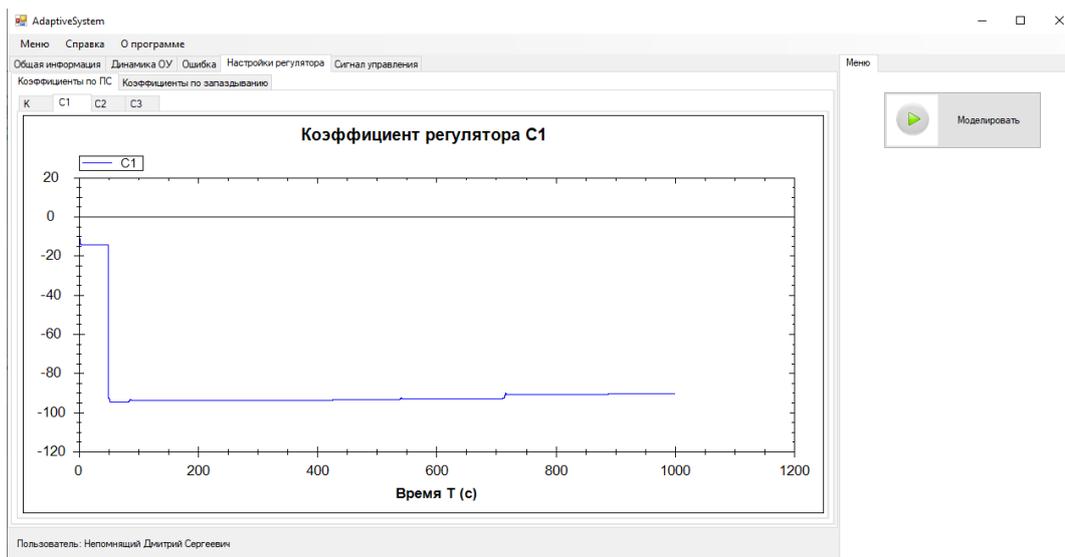


Рисунок 26 – Внешний вид окна «Настройки регулятора»

«Сигнал управления» - окно, отображающее формируемый в результате работы контура адаптации управляющее воздействие, подающегося на вход объекта управления для придания ему требуемой динамики. Внешний вид окна представлен на рисунке 27.

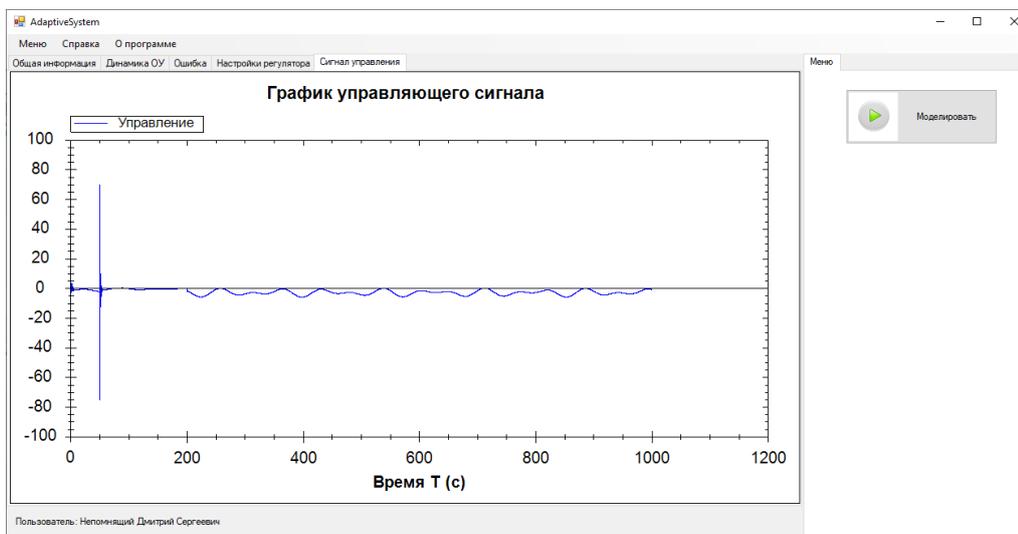


Рисунок 27 – Внешний вид окна «Сигнал управления»

Таким образом, после этапа моделирования, пользователь имеет все необходимые характеристики системы, а также мощный инструментарий для просмотра и работы с графиками.

## ЗАКЛЮЧЕНИЕ

В рамках выполнения магистерской диссертации были исследованы методы теории управления, позволяющие синтезировать адаптивные системы управления сложными объектами. Сложность объектов управления может заключаться в наличии запаздывания, наличии неопределенности различного вида, а также наличие объектов со сложной динамикой изменения параметров в процессе функционирования. Но, поскольку автоматизация технологических процессов продолжает усложняться, актуальность разработки новых методов и совершенствования традиционных продолжает расти.

В процессе синтеза адаптивной системы управления на основе критерия гиперустойчивости Попова, были получены алгоритмы настройки адаптивного регулятора, но ввиду сложности ОУ, данный алгоритм не позволял получить хорошее качество работы контура адаптации. При исследовании методов теории управления было принято решение о модификации алгоритмов контура адаптации, позволяющей исключать резкий рост ошибки во время переключения объекта на новые параметры. Данная модификация позволяет получить работоспособную систему только в случае одномоментного скачка параметров ОУ. При проведении имитационного моделирования синтезированной адаптивной системы управления была доказана работоспособность полученных алгоритмов.

После получения работоспособной системы управления была исследована необходимость разработки прикладного программного обеспечения для взаимодействия с системой. Ввиду сложности использования имеющихся средств и отсутствие возможностей автоматизации процесса настройки, расчета и получения характеристик системы была доказана необходимость прикладного ПО.

Разработанное прикладное программное обеспечение, построенное на основе клиент-серверной архитектуры, позволило использовать рабочие станции пользователей с малой вычислительной мощностью для работы с системой управления. Кроме того, прикладное ПО позволило автоматизировать процесс

Изм.	Лист	№ документа	Подпись	Дата

настройки и расчета параметров для имитационного моделирования, сохранять результаты моделирования для последующей обработки, использовать встроенные инструменты для работы с графиками для детального анализа.

Таким образом, поставленная цель и все задачи магистерской диссертации были достигнуты, что позволило закрепить и пополнить знания в области теории управления, а также разработке пользовательского прикладного программного обеспечения. В будущем разработанное прикладное программное обеспечение может быть модифицирована:

- добавлен модуль автоматической настройки коэффициентов адаптивного регулятора;
- добавлен модуль анализа полученных характеристик;
- добавлена возможность работать с объектами управления выше 3-го порядка.

Данные модификации позволят расширить область применения, разработанного в ходе магистерской диссертации, программного продукта.

## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- 1 Еремин, Е.Л. L-диссипативность гиперустойчивой системы управления при структурном возмущении / Е.Л. Еремин // Информатика и системы управления. – 2006. – № 2 (12). – С. 94-101.
- 2 Еремин, Е.Л. Алгоритмы адаптивной системы управления с явно-неявной эталонной моделью для строго минимально-фазового объекта / Е.Л. Еремин // Информатика и системы управления. – 2004. – № 2 (8). – С. 157-166.
- 3 Еремин, Е.Л. Построение адаптивных систем с запаздыванием по управлению на основе эталонного упредителя / Е.Л. Еремин // Информатика и системы управления. – 2005. – № 1 (9). – С. 122-128.
- 4 Еремин, Е.Л. Робастное управление нелинейными объектами с наблюдателем полного порядка и быстродействующей эталонной моделью / Е.Л. Еремин, Н.В. Кван, Н.П. Семичевская // Мехатроника, автоматизация, управление. – 2010. – № 5. – С. 2-6.
- 5 Еремин, Е.Л. Алгоритмы адаптивной системы с запаздыванием по управлению в схеме с расширенной ошибкой и эталонным упредителем / Е.Л. Еремин, Д.А. Теличенко // Мехатроника, автоматизация, управление. – 2006. – № 6. – С. 9-16.
- 6 Еремин, Е.Л. Периодические режимы в схемах децентрализованного адаптивного и робастного управления / Е.Л. Еремин, Д.А. Теличенко, Е.А.Шеленок // Вестник Рязанского государственного радиотехнического университета. – 2011. – № 35. – С. 108-116.
- 7 Еремин, Е.Л. Циклический режим в системе робастного управления манипулятором Барретта / Е.Л. Еремин, Д.А. Теличенко, Е.А. Шеленок // Вестник Тихоокеанского государственного университета. – 2010. – № 3. – С. 23-32.
- 8 Еремин, Е.Л. Синтез адаптивных систем управления на основе критерия гиперустойчивости / Е.Л. Еремин, А.М. Цыкунов. – Бишкек: Илим, 1992.

9 Еремин, Е.Л. Робастная система управления аффинным объектом в схеме с двумя эталонными моделями / Е.Л. Еремин, Л.В. Чепак // Информатика и системы управления. – 2014. – № 3 (41). – С. 121-129.

10 Еремин, Е.Л. Имитационное моделирование адаптивной системы управления одноканальным объектом с запаздыванием нейтрального типа и входным насыщением / Е.Л. Еремин, Е.А. Шеленок // Датчики и системы. – 2017. – № 10 (218). – С. 3-9.

11 Еремин, Е.Л. Робастные алгоритмы нестационарных систем управления с явно-неявной эталонной моделью / Е.Л. Еремин // Дифференциальные уравнения и процессы управления. – 2001. – № 3. – С. 61-74.

12 Еремин, Е.Л. Адаптивное управление динамических объектом на множестве состояний функционирования / Е.Л. Еремин // Информатика и системы управления. – 2012. - №4 (34) – С. 107 -118.

13 Ким, Д.П. Теория автоматического управления. Т. 1. Линейные системы / Д.П. Ким – Уч.Пос. – М.: ФИЗМАТЛИТ, 2016. – 312 с.

14 Рофэйл, Э. СОМ и СОМ+. Полное руководство / Э. Рофэйл, Я. Шохадуд – Уч.Пос. – М.: Корона Принт – 2009 – 560 с.

15 Никифоров, В.О. Интеллектуальное управление в условиях неопределенности / В.О. Никифоров, О.В. Слита, А.В. Ушаков. – Уч. Пос. – СПб.: СПбГУ ИТМО, 2011. – 226 с.

16 Дьяконов, В.П. МАТЛАВ. Полный самоучитель / В.П. Дьяконов // М.: Москва, ДМК Пресс, 2012. – 768с.

17 Культин, Н.Б. Основы программирования в Microsoft Visual C# 2010 / Н.Б. Культин // СПб.: БВХ – Петербург, 2011. – 368 с.

18 Троелсен, Э. Язык программирования С# 7 и платформы .NET и .NET Core / Э. Троелсен, Ф. Джепикс // 8-е изд.: Пер. с англ. – СПб. : ООО «Диалектика», 2020 – 1328с.

19 Албахари, Д С# 7.0 Справочник. Полное описание языка // Д. Албахари, Б. Албахари // пер. с англ. – СПб.: ООО «Диалектика», 2019. – 1024с.

20 ГОСТ 19.201-78 Единая система программной документации (УСПД).  
Техническое задание. Требования к содержанию и оформлению (с Изменением  
N 1). – М.: Изд-во Госстандарта России, 2010.

21 Логическое проектирование [Электронный ресурс]. – Режим доступа:  
<http://daxnow.narod.ru/index/0-16>. –18.04.2018

22 Реляционная модель данных [Электронный ресурс]. – Режим доступа:  
[http://www.bseu.by/it/tohod/lekci2\\_3.htm](http://www.bseu.by/it/tohod/lekci2_3.htm). –18.04.2018

23 Физическое проектирование [Электронный ресурс]. – Режим доступа:  
<http://bourabai.ru/dbt/dbms/03.htm>. –19.04.2018

24 HNB [Электронный ресурс]. – Режим доступа: <http://hnb.com.ua>. –  
20.04.2018

## ПРИЛОЖЕНИЕ А

### Построение эквивалентной системы

При последовательном соединении ОУ (27) и ДК (11), получим исходную систему, представленную на рисунке А.1.

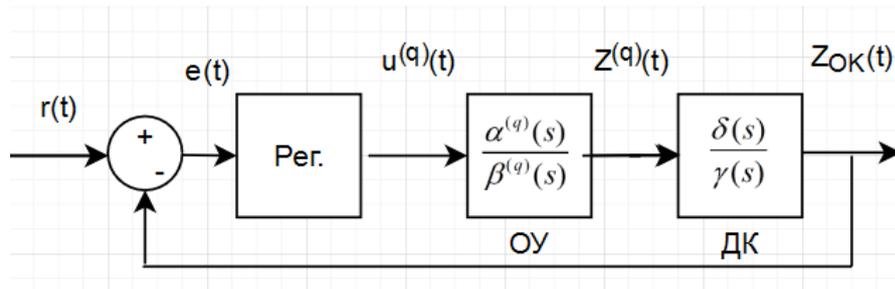


Рисунок А.1 – Структурная схема исходной системы

На рисунке А.1:  $r(t)$  – задающее воздействие;  $e(t)$  – ошибка, поступающая в регулятор;  $P$  – регулятор любой структуры;  $u^{(q)}(t)$  – управляющее воздействие, генерируемое в регуляторе для интервала времени  $q$ ; ОУ – объект управления; ДК – динамический корректор или фильтр-корректор;  $Z^{(q)}(t)$  – обобщенный выход объекта управления на интервале времени  $q$ ;  $Z^{(q)}_{ок}(t)$  – выход основного контура управления на интервале времени  $q$ .

В силу того, что объект управления и фильтр-корректор в структурной схеме соединены последовательно, возможно записать их общую передаточную функцию в виде:

$$\frac{\alpha^{(q)}(s) \delta(s)}{\beta^{(q)}(s) \gamma(s)} = \frac{1}{\gamma(s)} \frac{d(s)}{\beta^{(q)}(s)}, \quad (1)$$

где  $d(s)$  – гурвицев многочлен, степени  $p-1$ .

Имея общую передаточную функцию, вводятся следующие математические модели:

- модель видоизмененного объекта управления;
- модель видоизмененного динамического корректора.

Видоизмененный объект управления имеет вид:

## Продолжение ПРИЛОЖЕНИЯ А

$$z^{(q)}(s) = \frac{\delta(s)a^{(q)}(s)}{\beta^{(q)}(s)} u^{(q)}(s) = \frac{d(s)}{\beta^{(q)}(s)} u^{(q)}(s), \quad (2)$$

в свою очередь видоизменённый динамический корректор представлен уравнением:

$$\nu(s) = \frac{1}{\gamma(s)} u^{(q)}(s) = \frac{1}{(T_*s + 1)^2} u^{(q)}(s), \quad (3)$$

При введении моделей ВОУ и ВДК рассматриваются видоизмененная и упрощённая системы управления, представленные на рисунках А.2 и А.3.

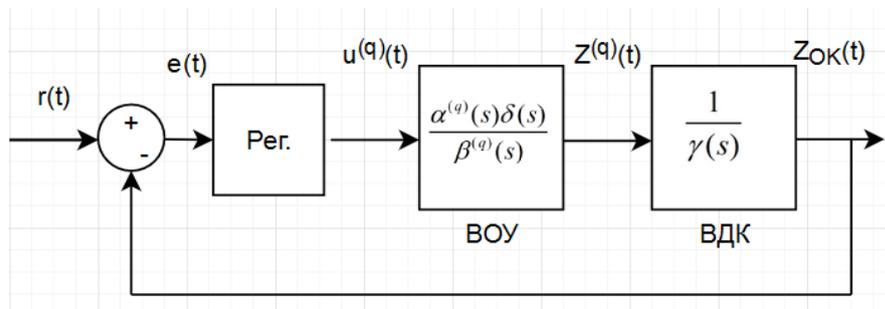


Рисунок А.2 – Видоизмененная система

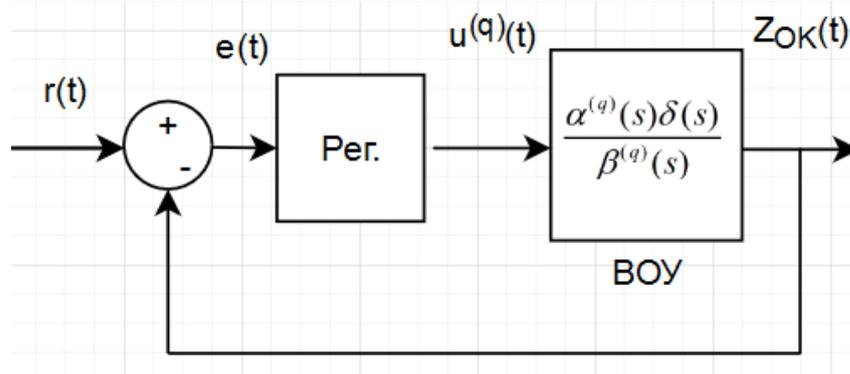


Рисунок А.3 – Упрощенная система

Видоизмененный динамический корректор, как показано в работе [6] возможно считать структурным возмущением и рассматривать упрощенную систему управления, представленную на рисунке А.3.

При этом при задании соответствующего значения постоянной времени  $T_*$  упрощенная система будет являться L-диссипативной, все траектории движения с течением времени будут входить в некоторое ограниченное множество. Для обеспечения условия L-диссипативности системы, представим ВОУ в виде:

Продолжение ПРИЛОЖЕНИЯ А

$$\beta_0 = (T_*^k s^k + \sum_{i=1}^{k-1} \theta_i T_*^{k-i} s^{k-i} + 1) + (d_0 s^{p-1} + \dots + d_{n-2} s + d_{n-1}), \quad (4)$$

где  $\theta$  – коэффициенты, определяемые соответствующим числом сочетаний:

$$\theta_1 = \theta_{k-1} = \frac{k!}{(k-1)!1!} = k, \theta_2 = \theta_{k-2} = \frac{k!}{(k-2)!2!}. \quad (5)$$

Если выбрать значение параметра  $T_* \ll 1$ , а правую часть многочлена (4) представить в виде:

$$\begin{aligned} \beta_0(s) = & T_*^k s^{k+p} + T_*^{k-1} (\theta_1 + T_* \beta_1) s^{k+p-1} + \dots + \\ & + T_*^2 (\theta_{k-2} + \theta_{k-3} T_* \beta_1 + \theta_{k-4} T_*^2 \beta_2 + \dots + \theta_1 T_*^{k-3} \beta_{k-3} + T_*^{k-2} \beta_{k-2}) s^{p+2} + \\ & + T_* (\theta_{k-1} + \theta_{k-2} T_* \beta_1 + \theta_{k-3} T_*^2 \beta_2 + \dots + \theta_1 T_*^{k-2} \beta_{k-2} + T_*^{k-1} \beta_{k-1}) s^{p+1} + \\ & + (1 + \theta_{k-1} T_* \beta_1 + \theta_{k-2} T_*^2 \beta_2 + \dots + \theta_1 T_*^{k-1} \beta_{k-1} + T_*^k \beta_k) s^p + \\ & + (d_0 + \beta_1 + \beta_2 \theta_{k-1} T_* + \beta_3 \theta_{k-2} T_*^2 + \dots) s^{n-1} + \\ & + (d_1 + \beta_2 + \beta_3 \theta_{k-1} T_* + \beta_4 \theta_{k-2} T_*^2 + \dots) s^{n-2} + \\ & + \dots + (d_{n-2} + \beta_{n-1} + \beta_3 \theta_{k-1} T_*) s + (d_{n-1} + \beta_n), \end{aligned} \quad (6)$$

то с высокой степенью приближения можно ввести замены:

$$\begin{aligned} T_*^{k-1} (\theta_1 + T_* \beta_1) & \cong T_*^{k-1} \theta_1, \dots, \\ T_*^2 (\theta_{k-2} + \theta_{k-3} T_* \beta_1 + \theta_{k-4} T_*^2 \beta_2 + \dots + \theta_1 T_*^{k-3} \beta_{k-3} + T_*^{k-2} \beta_{k-2}) & \cong T_*^2 \theta_{k-2}, \\ T_* (\theta_{k-1} + \theta_{k-2} T_* \beta_1 + \theta_{k-3} T_*^2 \beta_2 + \dots + \theta_1 T_*^{k-2} \beta_{k-2} + T_*^{k-1} \beta_{k-1}) & \cong T_* \theta_{k-1}, \\ (1 + \theta_{k-1} T_* \beta_1 + \theta_{k-2} T_*^2 \beta_2 + \dots + \theta_1 T_*^{k-1} \beta_{k-1} + T_*^k \beta_k) & \cong 1, \\ (d_0 + \beta_1 + \beta_2 \theta_{k-1} T_* + \beta_3 \theta_{k-2} T_*^2 + \dots) & \cong d_0 \\ (d_1 + \beta_2 + \beta_3 \theta_{k-1} T_* + \beta_4 \theta_{k-2} T_*^2 + \dots) & \cong d_1 \\ (d_{p-1} + \beta_p) & \cong d_{p-1}. \end{aligned} \quad (7)$$

Используя соотношения (6) и (7) перепишем (4) в виде:

$$\begin{aligned} \beta_0(s) = & T_*^k s^{k+p} + T_*^{k-1} \theta_1 s^{k+p-1} + \dots + T_*^2 \theta_{k-2} s^{p+2} + \\ & + T_* \theta_{k-1} s^{p+1} + s^p + d_0 s^{p-1} + d_1 s^{p-2} + \dots + d_{p-1} \end{aligned} \quad (8)$$

Таким образом, для обеспечения условия L-диссипативности объекта (27), требуется выполнить условие утверждения, показанного в работе [6]:

$$\begin{aligned} 0 < T_* < T_*^+, \\ (T_*^+)_1 & \leq \frac{0.93}{(k-1)d_1}, (T_*^+)_2 \leq \frac{0.465d_1}{kd_2}, \end{aligned} \quad (9)$$

## ПРИЛОЖЕНИЕ Б

Simulink – схема системы при имитационном моделировании

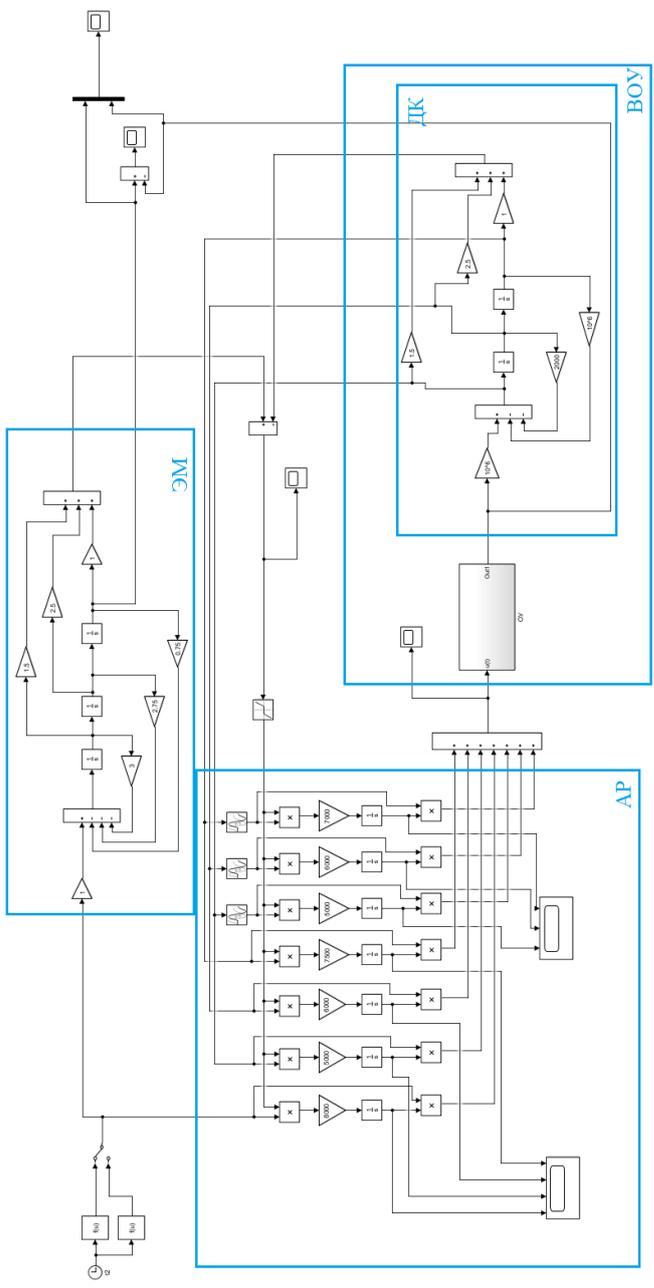


Рисунок Б.1 – Simulink – схема

# ПРИЛОЖЕНИЕ В

## Основные этапы разработки спиральной модели

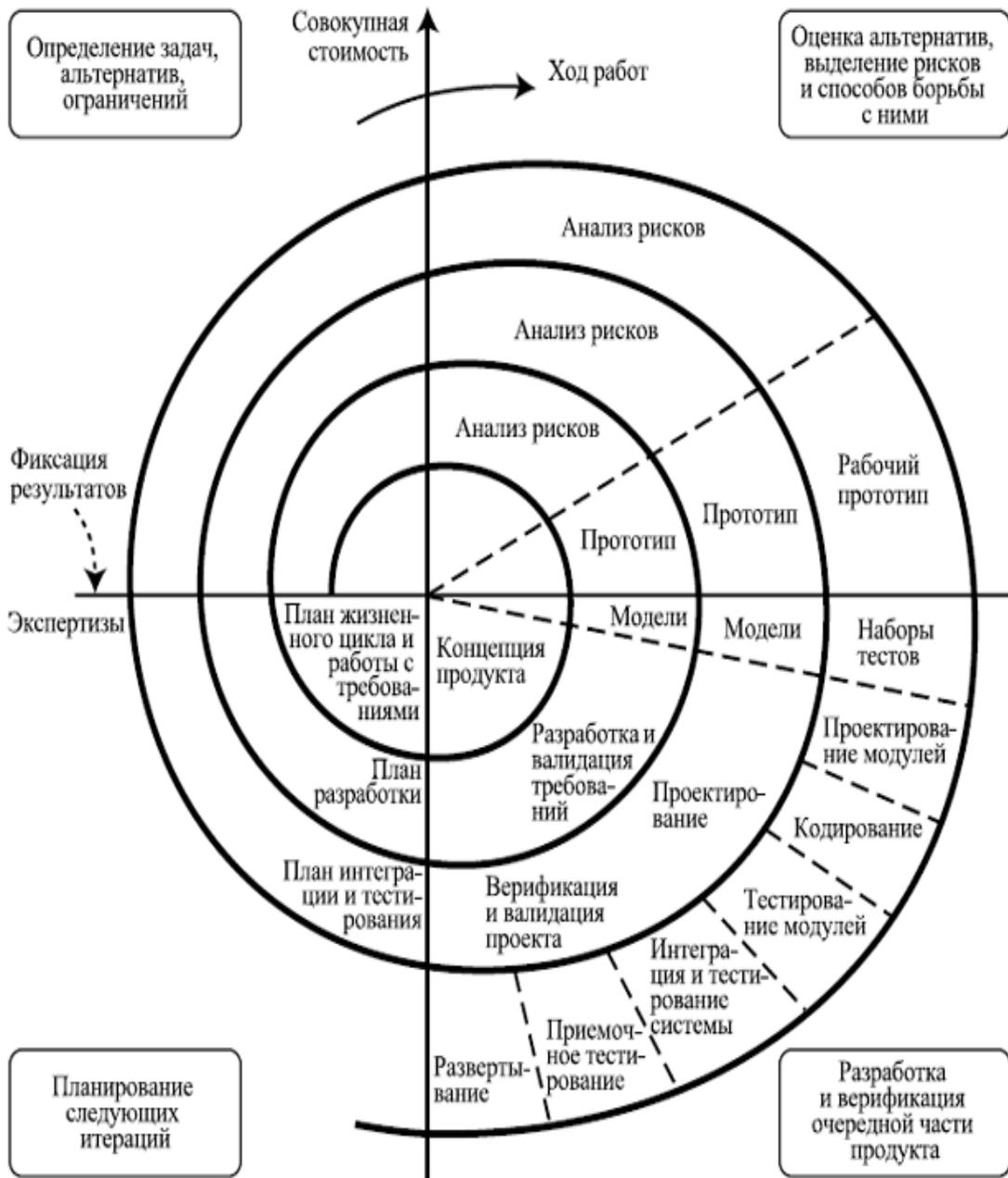


Рисунок В1 – Основные этапы разработки спиральной модели

# ПРИЛОЖЕНИЕ Г

## Функциональная диаграмма прикладного программного обеспечения

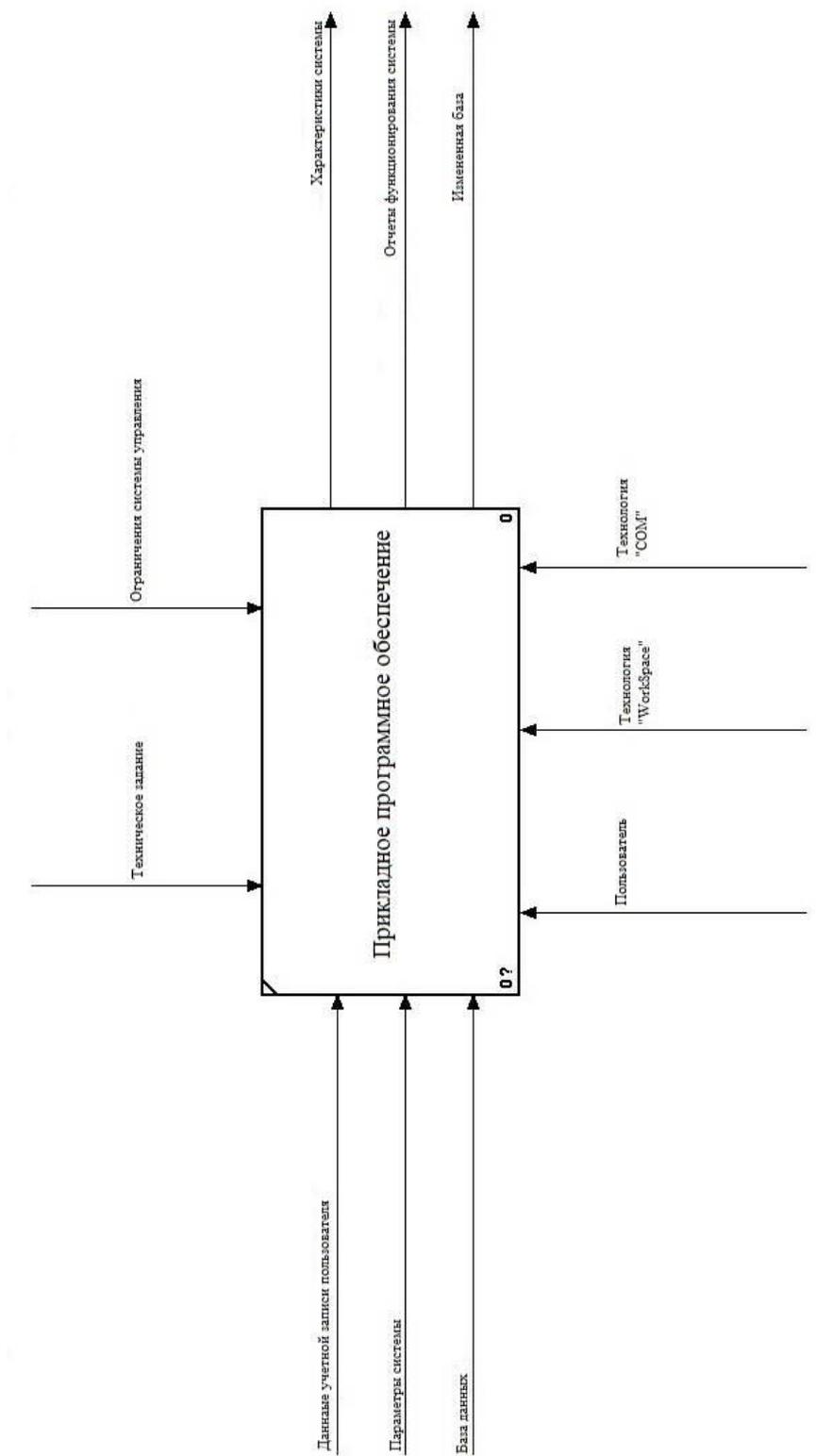


Рисунок Г1 – Функциональная диаграмма

# ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ Г

## Декомпозиция функциональной диаграммы

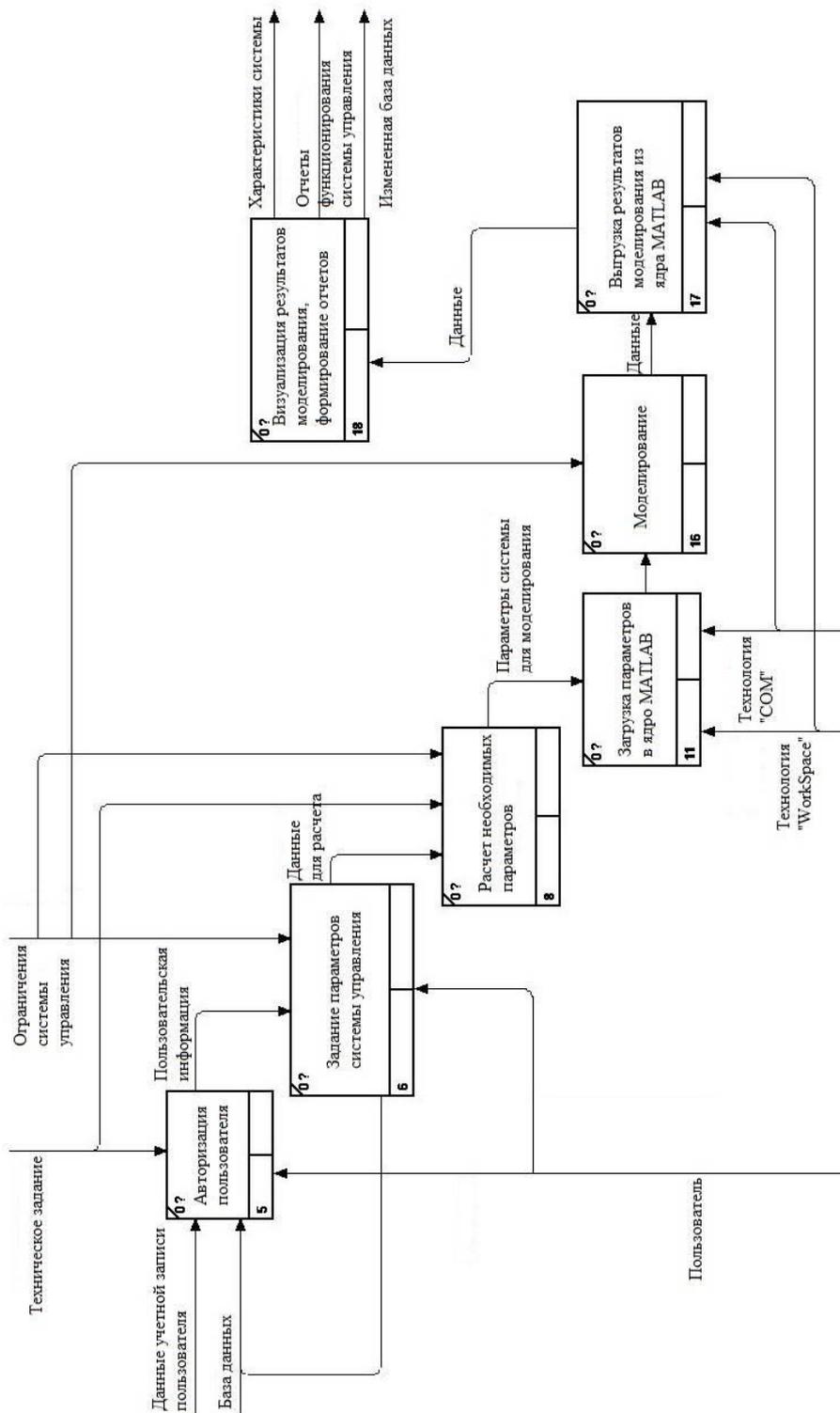


Рисунок Г2 – Декомпозиция функциональной диаграммы

## ПРИЛОЖЕНИЕ Д

### UML-диаграммы разрабатываемого программного обеспечения

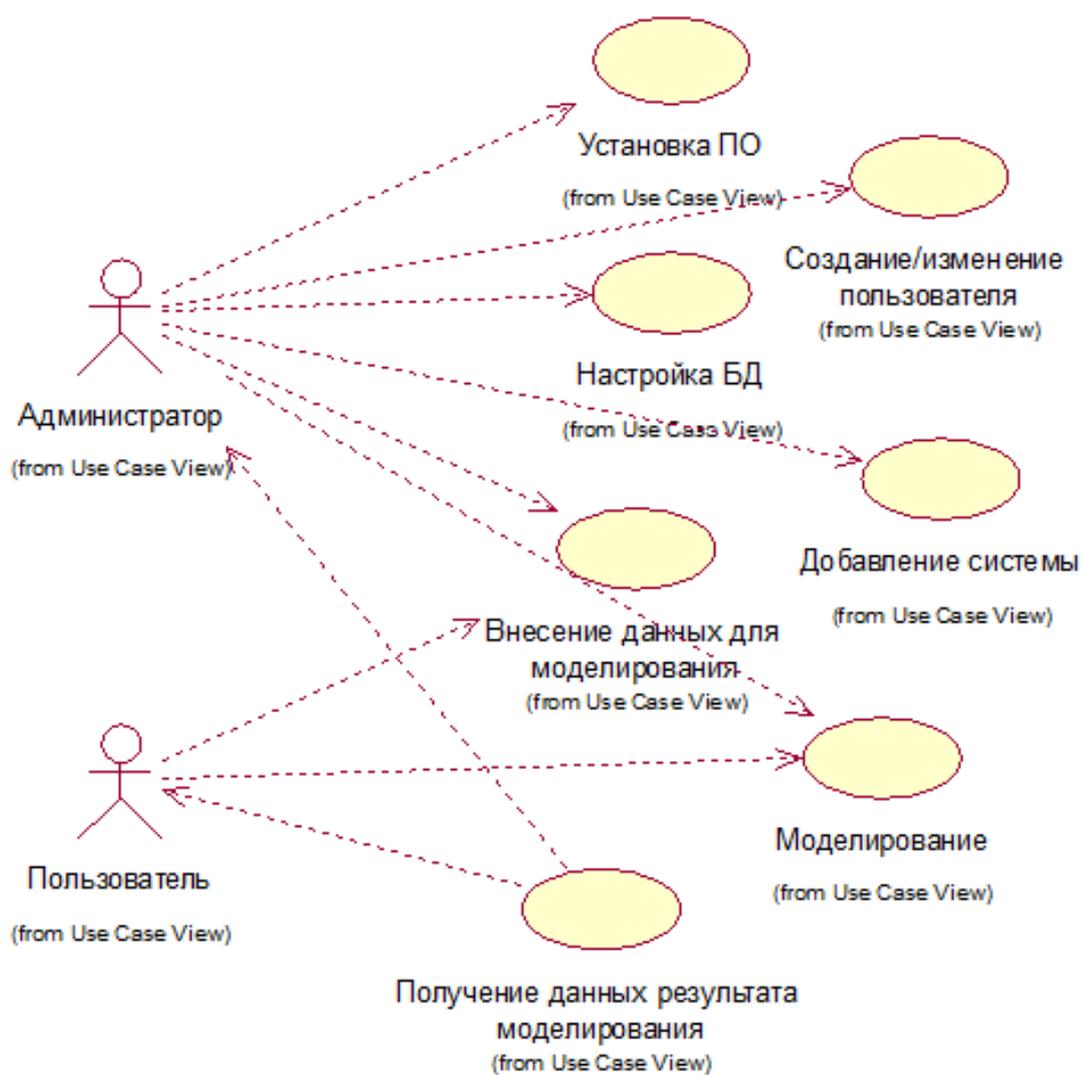


Рисунок Д1 – Диаграмма вариантов использования

# ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ Д

## Диаграмма последовательности

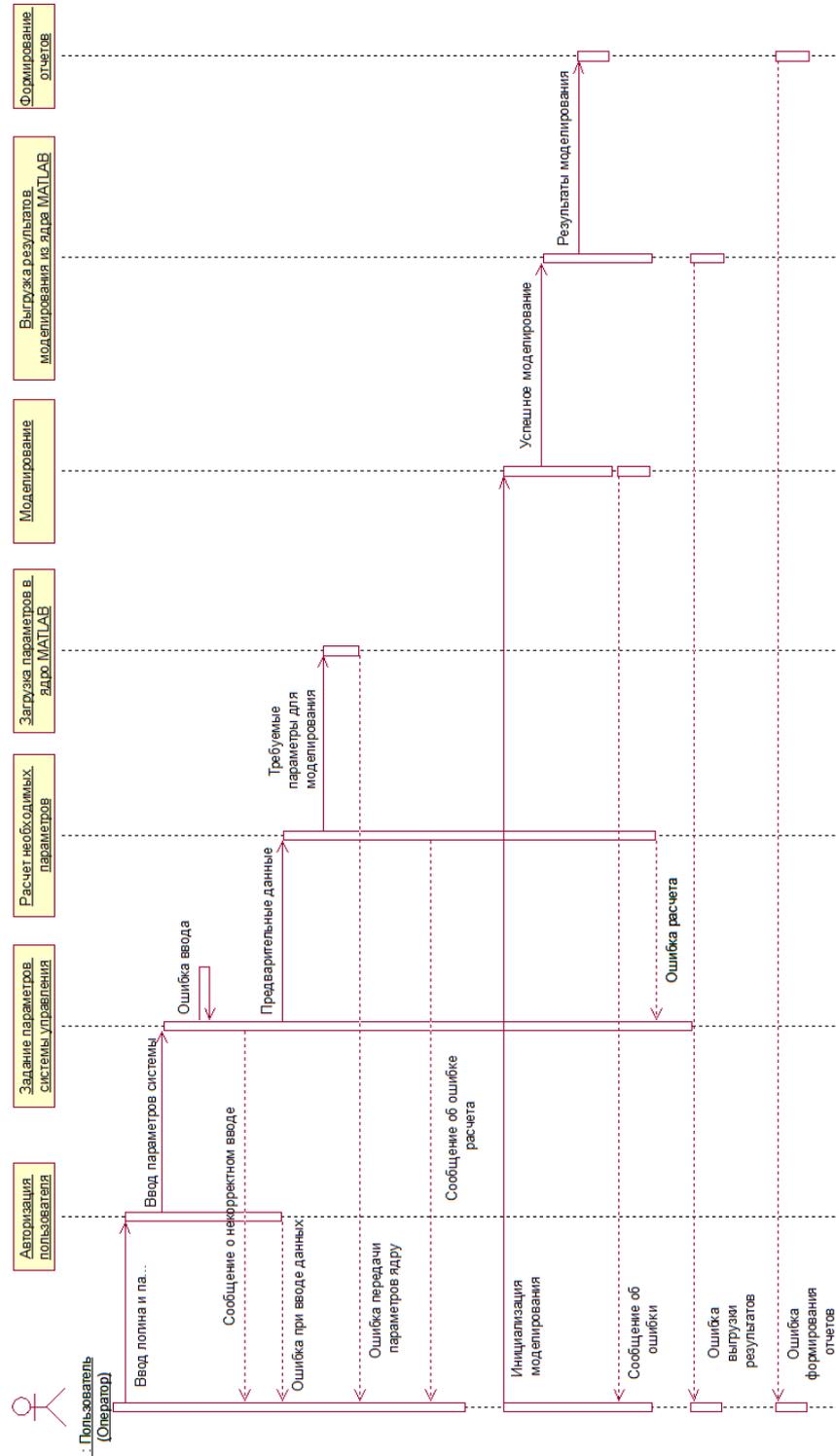


Рисунок Д2 – Диаграмма последовательности

# ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ Д

## Диаграмма состояний

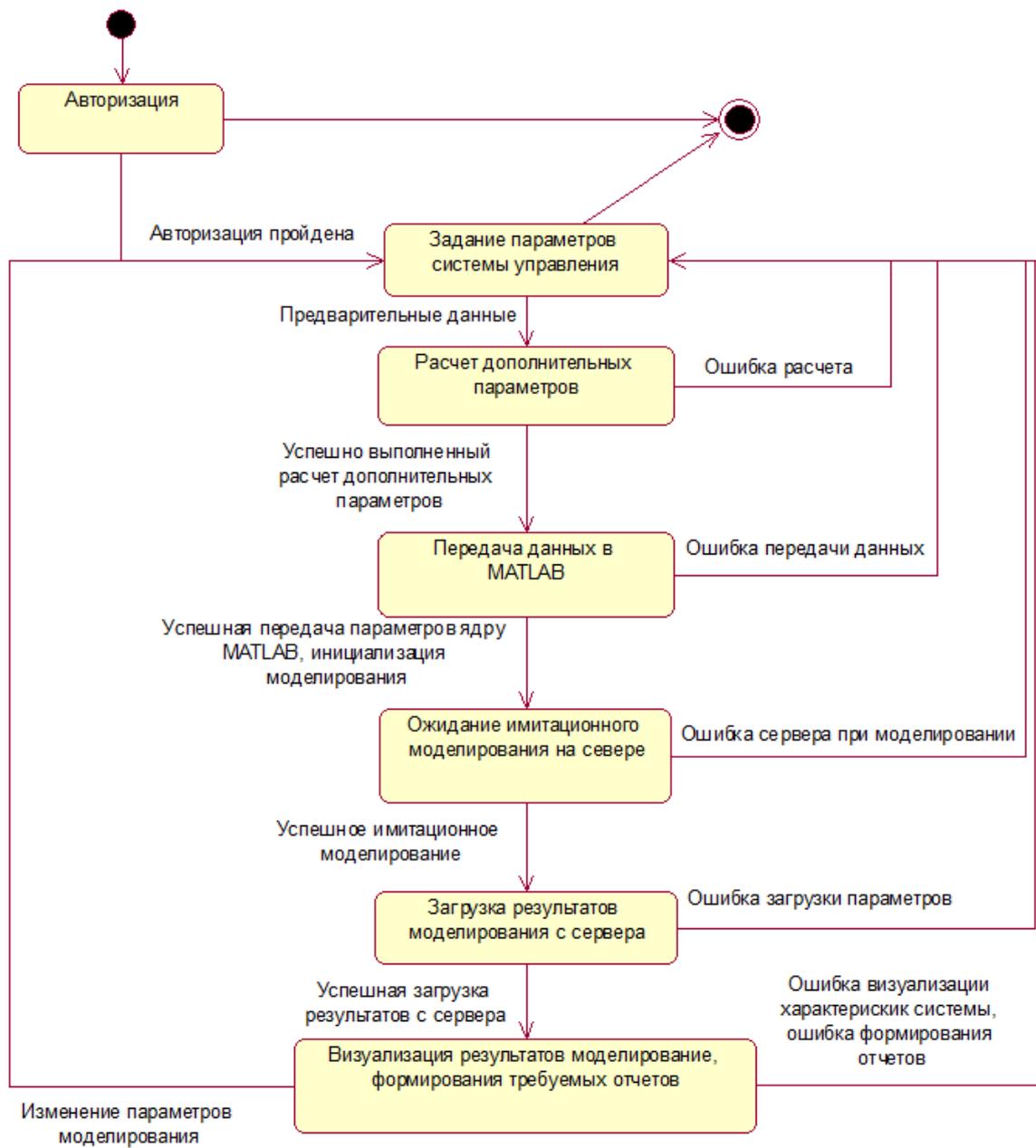


Рисунок Д3 – Диаграмма состояний

# ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ Д

## Диаграмма активности

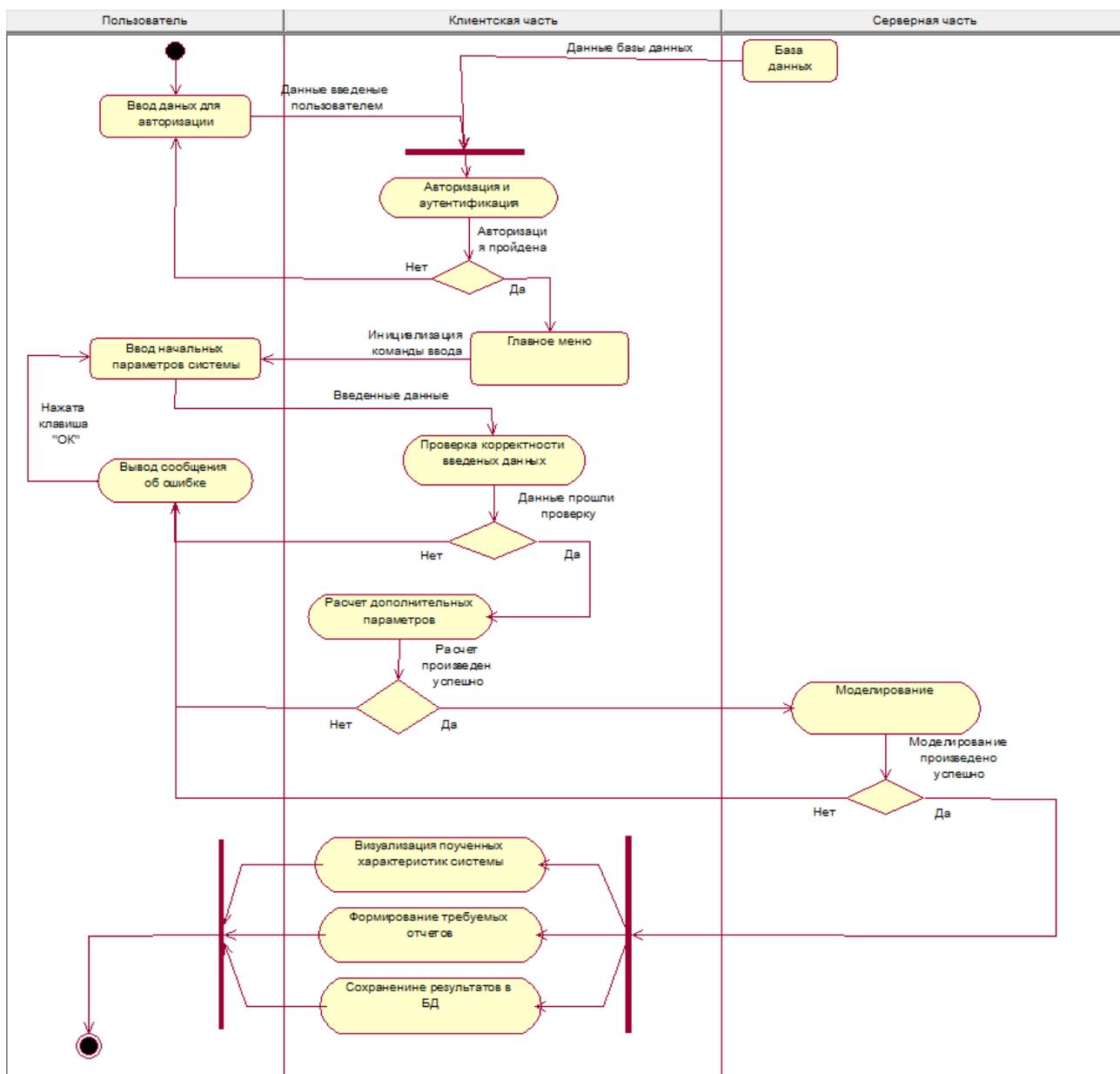


Рисунок Д4 – Диаграмма активности

# ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ Д

## Диаграмма классов

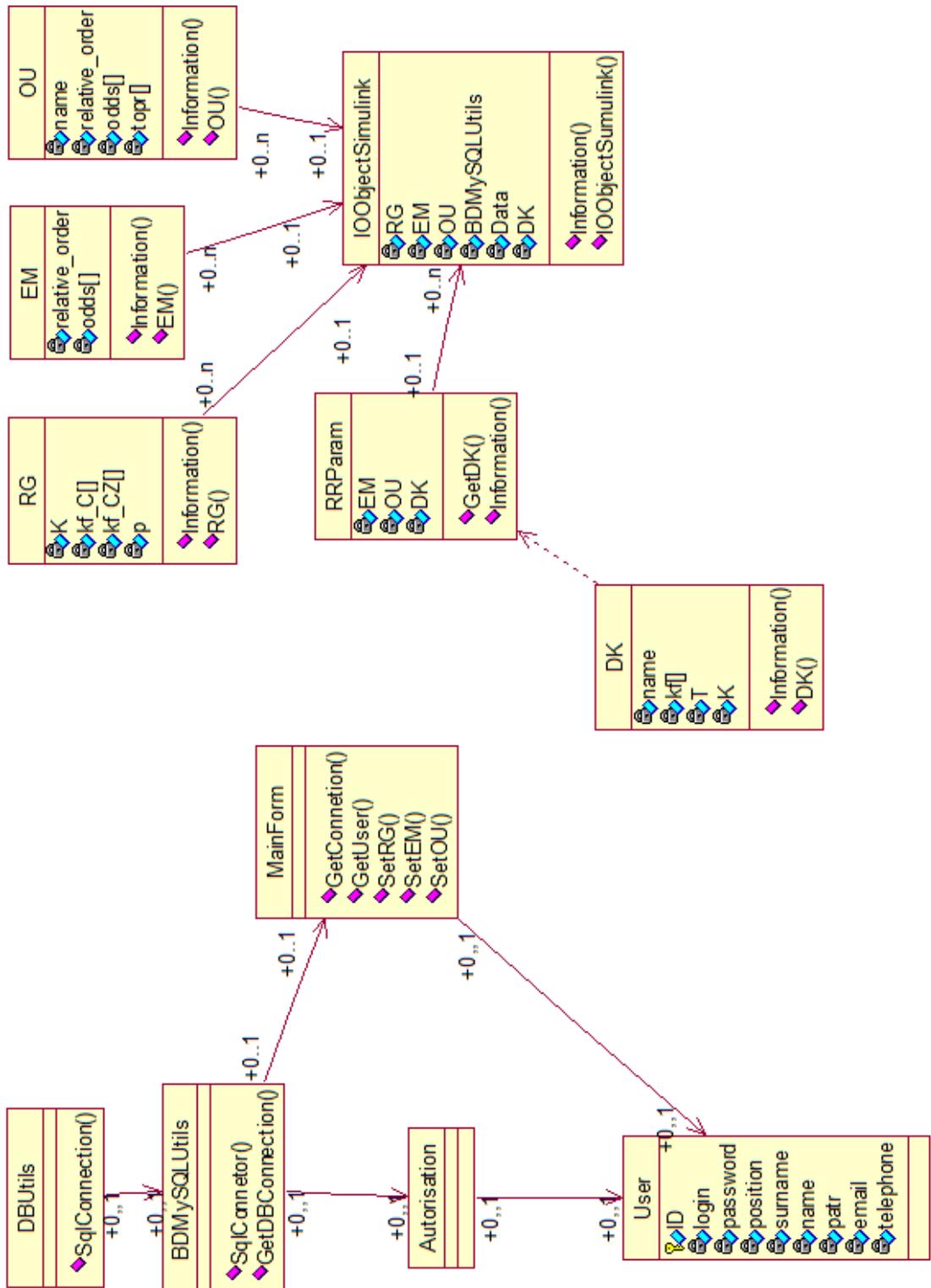


Рисунок Д5 – Диаграмма классов

## ПРИЛОЖЕНИЕ Е

### Блок схема алгоритма расчета имитационных характеристик

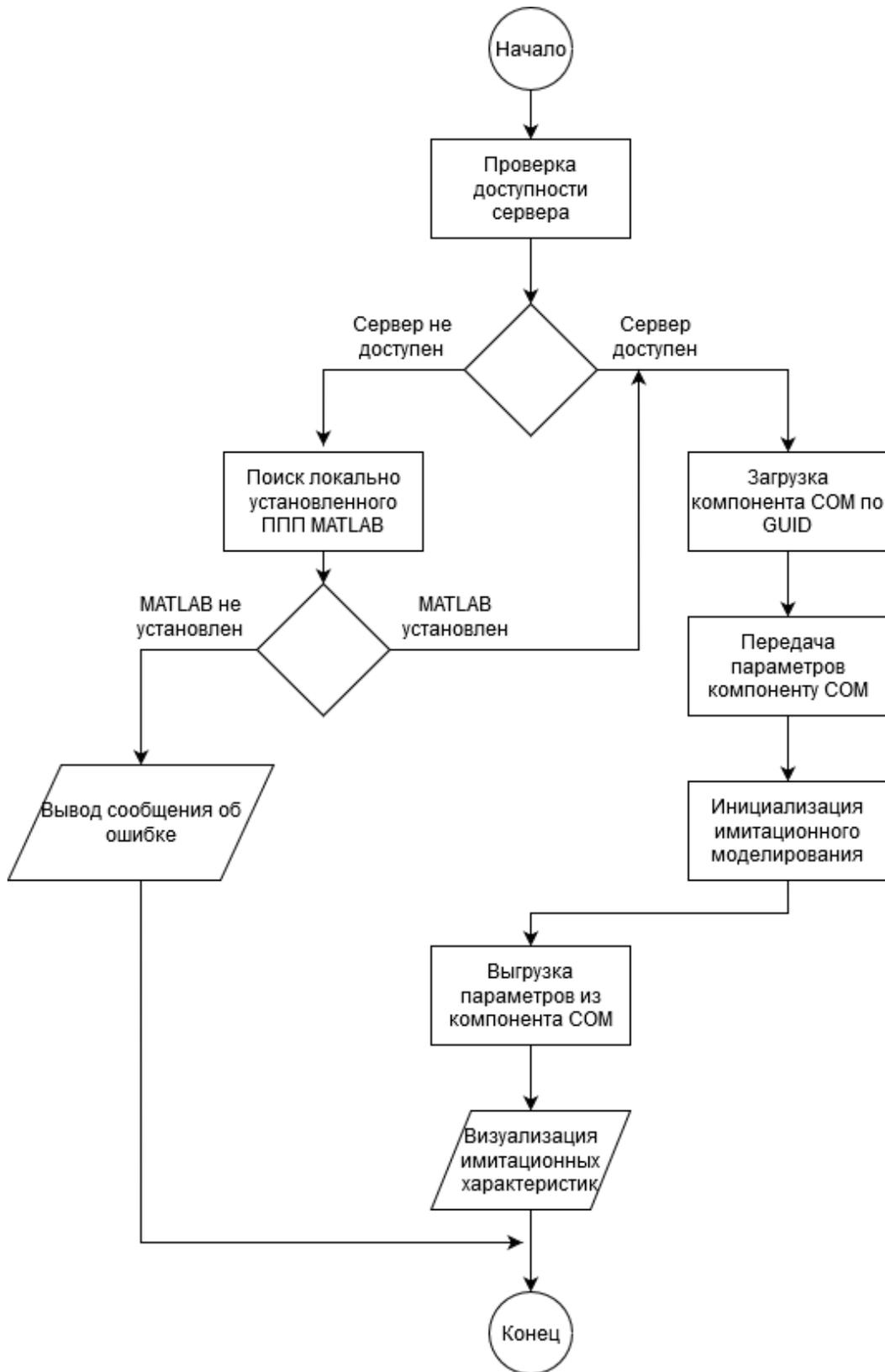


Рисунок Е1 – Блок схема расчета имитационных характеристик