МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение высшего образования   
«САНКТ–ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ   
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

ДОПУСТИТЬ К ЗАЩИТЕ

Заведующий кафедрой №43

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| д.т.н, профессор |  |  |  | М.Ю. Охтилев |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

БАКАЛАВРСКАЯ РАБОТА

|  |  |
| --- | --- |
| на тему | Моделирование пополнения однопродуктового склада в условиях |
| стохастической неопределённости | |
|  | |

|  |  |
| --- | --- |
| выполнена | Соколовой Екатериной Юрьевной |
| фамилия, имя, отчество студента в творительном падеже | |

|  |  |  |  |
| --- | --- | --- | --- |
| по направлению подготовки | 09.03.04 |  | Программная инженерия |
|  | код |  | наименование направления |
|  | | | |
| наименование направления | | | |
| направленности |  |  | Проектирование программных систем |
|  | код |  | наименование направленности |
|  | | | |
| наименование направленности | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Студент группы № | 4931 |  |  |  | Соколова Е.Ю. |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Руководитель

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| д.т.н., доцент |  |  |  | Колесникова С.И. |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2023

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение высшего образования   
«САНКТ–ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ   
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

УТВЕРЖДАЮ

Заведующий кафедрой №43

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| д.т.н, профессор |  |  |  | М.Ю. Охтилев |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

ЗАДАНИЕ НА ВЫПОЛНЕНИЕ БАКАЛАВРСКОЙ РАБОТЫ

|  |  |  |  |
| --- | --- | --- | --- |
| студенту группы | 4931 |  | Соколовой Екатерине Юрьевне |
|  | номер |  | фамилия, имя, отчество |

|  |  |
| --- | --- |
| на тему | Моделирование пополнения однопродуктового склада в условиях |
| стохастической неопределённости | |
|  | |

|  |  |  |  |
| --- | --- | --- | --- |
| утверждённую приказом ГУАП от |  | № |  |

|  |  |
| --- | --- |
| Цель работы: | исследование целесообразности использования метода аналитического |
| конструирования агрегированных регуляторов для оптимизации однопродуктового склада | |
|  | |

|  |  |
| --- | --- |
| Задачи, подлежащие решению: | 1. Обзор метода аналитического конструирования |
| агрегированных регуляторов; 2. Получение формул для управления; 3. Разработка | |
| программного обеспечения для тестирования полученной модели. | |

|  |  |
| --- | --- |
| Содержание работы (основные разделы): | 1. Теоретические основы; 2. Постановка |
| задачи управления; 3. Аналитический синтез регуляторов; 4. Описание разработанной | |
| программы; 5. Тестирование | |
|  | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Срок сдачи работы « | 31 | » | 05 | 20 | 23 |

Руководитель

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| д.т.н., доцент |  |  |  | Колесникова С.И. |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

Задание принял(а) к исполнению

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| студент группы № | 4931 |  |  |  | Соколова Е. Ю. |
|  |  |  | подпись, дата |  | инициалы, фамилия |

**РЕФЕРАТ**

Выпускная квалификационная работа 110 стр., 72 рис., 11 табл., 21 источник., 1 приложение.

Актуальность: любые объекты, математическая модель которых включает случайную составляющую, не имеют однозначного решения. Поэтому любые алгоритмы, которые позволяют аналитически дать рекомендации по заполнению склада в условии возмущения и шумов актуальны.

Цель работы: исследование целесообразности использования метода аналитического конструирования агрегированных регуляторов (АКАР) для оптимизации однопродуктового склада.

Задачи, реализующие цель исследования:

1. обзор метода аналитического конструирования агрегированных регуляторов (АКАР);
2. получение формул для управления на основе обозреваемого метода;
3. разработка программного обеспечения для тестирования полученной модели и её вариаций;
4. исследование зависимости модели от параметра регулятора и исходных данных.

Основные результаты:

1. Система уравнений для модели однопродуктового склада;
2. Приложение для тестирования и анализа сконструированной системы управления.

Теоретическая значимость: новая система управления для модели управления однопродуктового склада в условиях стохастической неопределённости.

Практическая значимость: приложение для проведения численных экспериментов с моделью, позволяющее пользователю для исходных данных спрогнозировать частотность и объём поставок.

**REPORT**

Final qualifying work 110 pages, 72 figures, 11 tables, 21 sources, 1 appendix.

Relevance: any objects which mathematical model includes a random component do not have an unambiguous solution. Therefore, any algorithms, which allow you to give analytical recommendations on fulfilling the warehouse under the condition of random disturbance and noise, are relevant.

The purpose of the work: to study the feasibility of using the method of analytical design of aggregated regulators (ADAR) to optimize a single-product warehouse.

Tasks that implement the purpose of the study:

1. overview of the method of analytical design of aggregated regulators (ADAR);

2. obtaining formulas for control based on the method being reviewed;

3. software development for testing the resulting model and its variations;

4. investigation of the dependence of the model on the parameter of the regulator and the initial data.

Main results:

1. A system of equations for a single-product warehouse model;

2. Application for testing and analysis of the designed control system.

Theoretical significance: a new control system for the management model of a single-product warehouse under conditions of stochastic uncertainty.

Practical significance: an application for conducting numerical experiments with a model that allows users to predict frequency and volume of deliveries for the initial data.

**СОДЕРЖАНИЕ**

[ПЕРЕЧЕНЬ ОБОЗНАЧЕНИЙ И СОКРАЩЕНИЙ 5](#_Toc136381334)

[ВВЕДЕНИЕ 6](#_Toc136381335)

[ГЛАВА 1. ТЕОРЕТИЧЕСКИЕ ОСНОВЫ 9](#_Toc136381336)

[1.1 Основные понятия теории запасов 9](#_Toc136381337)

[1.2 Основные положения метода АКАР 13](#_Toc136381338)

[ГЛАВА 2. ПОСТАНОВКА ЗАДАЧИ УПРАВЛЕНИЯ 15](#_Toc136381339)

[2.1 Постановка задачи управления запасами однопродуктового слада 15](#_Toc136381340)

[ГЛАВА 3. АНАЛИТИЧЕСКИЙ СИНТЕЗ РЕГУЛЯТОРОВ 17](#_Toc136381341)

[3.1 Синтез управления запасами на основе метода АКАР 17](#_Toc136381342)

[ГЛАВА 4. ОПИСАНИЕ РАЗРАБОТАННОЙ ПРОГРАММЫ 20](#_Toc136381343)

[4.1 Функциональные требования к программе 20](#_Toc136381344)

[4.2 Требования к используемому программному обеспечению 20](#_Toc136381345)

[4.3 Алгоритм работы программы 21](#_Toc136381346)

[4.4 Графический интерфейс 25](#_Toc136381347)

[ГЛАВА 5. ТЕСТИРОВАНИЕ 27](#_Toc136381348)

[5.1 Моделирование системы управления методом АКАР 27](#_Toc136381349)

[5.2 Оценка качества полученных моделей 50](#_Toc136381350)

[5.3 Анализ влияния параметра регулятора на модель 73](#_Toc136381351)

[5.4 Анализ влияния начальных условия на модель 75](#_Toc136381352)

[5.5 Итоговая таблица 78](#_Toc136381353)

[ЗАКЛЮЧЕНИЕ 79](#_Toc136381354)

[СПИСОК ЛИТЕРАТУРЫ 80](#_Toc136381355)

[ПРИЛОЖЕНИЕ 83](#_Toc136381356)

# **ПЕРЕЧЕНЬ ОБОЗНАЧЕНИЙ И СОКРАЩЕНИЙ**

АКАР–аналитическое конструирование агрегированных регуляторов.

ПО –программное обеспечение.

ОУ – объект управления.

ИТ – изображающая точка.

СОФ – сопровождающий обобщённый функционал.

ОШС – отношение шум/сигнал.

MVP (minimal viable product) – минимально жизнеспособный продукт.

# **ВВЕДЕНИЕ**

Возможно, вы хоть раз сталкивались с тем, что, придя в магазин, вы обнаруживали отсутствие необходимого товара. Это может свидетельствовать о неправильном управлении запасами на складе, которое привело к дефициту товара и убыткам предприятия. Для избежание дефицита, прогнозирования спроса и оптимизации работы применяются математические модели управления запасами.

Теория управления запасами относится к числу наиболее молодых отраслей исследования. Впервые подобная задача применительно к определению резервных денежных фондов была математически сформулирована Эджвортом Ф. в 1888 г. В начале XX века появился целый ряд статей по определению наиболее экономичного объёма поставки марки материального ресурса на предприятие. Основы современной теории управления запасами – постановка задачи, анализ влияющих на решение факторов, способ учёта неопределённости в спросе – были сформулированы в работах Эрроу К., Гарриса В., Маршака С. и Дворецкого А.

Управлять запасами необходимо не только розничным магазина. На самом деле запасы пронизывают весь деловой мир. Поддержание товарных запасов необходимо любой компании, имеющей дело с товарами, включая производителей, оптовых и розничных продавцов. Например, производителям  нужны запасы материалов, необходимых для изготовления их продукции. Им также нужны запасы готовой продукции, ожидающей отгрузки. Аналогичным образом, как оптовым продавцам, так и розничным торговцам необходимо поддерживать запасы товаров, чтобы они были доступны для покупки. Однако, хранение товаров имеет свою цену, поэтому необходимо учитывать их срок годности, затраты на хранение (аренда помещения, электричество, охрана) и т.д.

Для того чтобы учесть все нюансы и составить оптимальный план, используются математические модели теории запасов, которые обеспечивают мощный инструмент для получения конкурентных преимуществ. Использование теории запасов компаниями включает в себя следующие этапы.

1. Формулировка математической модели, описывающей поведение системы запасов;
2. Поиск оптимального плана хранения и заказа товаров в отношении этой модели;
3. Использование компьютера для обработки математической модели, обработки информации, ведения учёта текущих запасов;
4. Построение оптимального плана пополнения запасов.

Математические модели теории запасов можно разделить на две широкие категории: детерминированные и стохастические в зависимости от спроса. Спрос – это количество единиц товара, которое необходимо изъять из запасов для некоторого использования в течении определённого промежутка времени. Если спрос может быть спрогнозирован со значительной точностью, используется детерминированная модель. Однако, когда величина спроса точно неизвестна, возникает необходимость использования стохастической модели запасов, в которой спрос в любой период является случайной величиной, а не известной константой.

В настоящее время есть множество моделей управления запасами, на основе которых проводятся исследования и создаются приложения, однако, в силу высокой неопределённости, связанной со спросом, временем выполнения заказа, непредвиденными обстоятельствами (забастовка сотрудников, задержка поставки, порча груза) возникает необходимость совершенствования алгоритмов на основе использования аппарата стохастической теории управления запасами.

Основываясь на вышесказанном, в выпускной квалификационной работе ставятся следующие задачи:

* изучение метода аналитического конструирования агрегированных регуляторов (АКАР), который используется для моделирования однопродуктового склада в условиях стохастической неопределённости;
* получение формул для управления на основе выбранного метода;
* разработка тестового программного обеспечения (ПО), которое будет иметь необходимый интерфейс для работы с моделью и являться базой для будущего сопровождения;
* исследование качества модели в зависимости от её параметров;
* тестирование модели, с целью выявления более эффективного её вида.

Приложение, которое будет получено в ходе выполнения дипломной работы, может позиционировать себя, как минимально жизнеспособный продукт (MVP), который направлен в первую очередь на тестирование выбранной модели, исследования её качества и работоспособности.

# **ГЛАВА 1. ТЕОРЕТИЧЕСКИЕ ОСНОВЫ**

## 1.1 Основные понятия теории запасов

Слово «запас» означает количество ресурсов любого рода, имеющих экономическую ценность. Другими словами, запас это некоторый объём ресурсов, который хранится под рукой для бесперебойного и эффективного ведения будущих дел организации. Это может быть сырье, товары на этапе производства, запасные части/расходные материалы, готовая продукция, человеческие ресурсы, такие как рабочая сила, финансовые ресурсы, такие как оборотный капитал и т.д. Необязательно, чтобы у организации были все эти классы запасов, но какими бы ни были товарно-материальные запасы, они нуждаются в эффективном управлении, поскольку, как правило, в них инвестируется значительная сумма денег. Теория запасов решает три основные задачи: 1) сколько заказать; 2) когда делать заказ; 3) какой объем страхового запаса следует хранить.

Проблемы, с которыми сталкиваются различные организации, потребовали использования научных методов в управлении товарно-материальными запасами, известных как контроль товарно-материальных запасов. Управление запасами связано с приобретением, хранением и обработкой товарно-материальных запасов таким образом, чтобы они были доступны в любое время, когда это необходимо, и связанные с этим общие затраты были сведены к минимуму.

Основные цели теории запасов:

1. улучшение обслуживания клиентов – способность удовлетворить запрос потребителя в любой момент времени;
2. сокращение затрат (расходы на хранение или транспортировку) – это расходы, связанные с хранением и доставкой товаров. Однако, хранение запасов на складе может косвенно снизить эксплуатационные расходы, такие как потеря деловой репутации и/или потеря потенциальных продаж из-за нехватки товара. Что может способствовать вложению сэкономленных денег в производство, для выпуска более крупных партий товара;
3. поддержание работоспособности – запасы сырья и товаров в процессе создания служат буфером между последовательными этапами производства, чтобы простои на одном из этапов не влияли на весь производственный процесс;
4. нерегулярное предложение и спрос – запасы обеспечивают защиту от нерегулярного предложения и спроса; неожиданное изменение графика производства или поставки товара может отрицательно сказаться на эксплуатационных расходах и уровне обслуживания клиентов;
5. скидка на количество – заказы большого размера помогают воспользоваться скидкой за объём. Однако такое преимущество должно обеспечивать баланс между стоимостью хранения и затратами, связанными со сроком годности, повреждением, кражей, страхованием и т.д.;
6. предотвращение перебоев поставок (дефицита) – в таких ситуациях как забастовки работников, колебания спроса, стихийные бедствия и задержки с поставками запасы выступают в качестве страховки против истощения запасов, а также потери деловой репутации.

Классификация затрат, связанных с товарно-материальным запасом, может быть представлена следующими положениями.

1. Стоимость покупки (или производства): это стоимость, по которой приобретается или производится товар;
2. Затраты на хранение: затраты, связанные с поддержанием товарно-материальных запасов, известны как затраты на хранение. Они прямо пропорциональны количеству товара на складе и времени, в течение которого товар хранится на складе. Сюда входят расходы на погрузочно-разгрузочные работы, техническое обслуживание, амортизацию, страхование, аренду склада, налоги и т.д.;
3. Стоимость нехватки (или отсутствия товара на складе): это стоимость, которая возникает из-за нехватки товара на складе. Сюда входят затраты на остановку производства, потерею деловой репутации, снижение рентабельности, специальные заказы по более высокой цене, выплаты за сверхурочную работу/простой, потерю возможности продавать и т.д.;
4. Стоимость заказа: стоимость, понесённая при пополнении запасов, известна как стоимость заказа. Сюда входят все расходы, связанные с администрированием (например, заработная плата сотрудников, занимающихся закупками, телефонные звонки, расходы на компьютер, почтовые расходы и т.д.), транспортировка, получение и досмотр товаров, обработка платежей и т.д. Если фирма производит свои собственные товары вместо того, чтобы закупать их у внешнего источника, то это затраты на перенастройку оборудования для производства.

В теории запасов используются следующие основные термины:

* спрос – это желание, которое связано с определенным временем, ценой и количеством. Структура спроса на товар может быть как детерминированной, так и вероятностной. В случае детерминированного спроса объёмы, необходимые в будущем, известны с определённость. Это может быть фиксированным (статическим) или может изменяться (динамическим) время от времени. Напротив, вероятностный спрос неопределёнен в течение определенного периода времени, но его структура может быть описана известным распределением вероятностей;
* график пополнения запасов – определяется, как период времени между двумя последовательными пополнениями. Заказ может быть сделан на основе двух типов систем проверки заказов: 1) непрерывный контроль – в этом случае уровень запасов отслеживается непрерывно до тех пор, пока не будет достигнута указанная точка (известная, как точка повторного заказа), на этом этапе оформляется новый заказ; 2) периодическая проверка – в этом случае заказы формируются через равные промежутки времени, количество, заказываемое каждый раз, зависит от уровня запасов на момент проверки;
* период планирования – известный также как временной горизонт, в течение которого необходимо контролировать уровень запасов. Он может быть конечным или бесконечным в зависимости от характера спроса;
* время выполнения заказа или задержка с доставкой – промежуток времени между моментом формирования заказа и фактическим его получением. Время выполнения заказа может быть детерминированным (постоянным или переменным) или вероятностным;
* страховой запас: как правило, спрос и сроки выполнения заказа не определены и не могут быть полностью предугаданы. Таким образом, чтобы учесть колебания спроса и предложения, сохраняется некоторый дополнительный запас. Этот запас известен также как страховой;
* величина повторного заказа: уровень между максимальными и минимальными запасами, при котором должна начаться закупочная деятельность для пополнения, называется уровнем повторного заказа.

В данной работе рассматривается стохастическая линейная модель запасов и линейная модель спроса, что возможно, например, при нестабильной работе поставщиков или нехватке транспортных средств.

## 1.2 Основные положения метода АКАР

Пусть объект управления описывается векторно-матричным дифференциальным уравнением

(1.2.1)

где векторы фазовых координат и управления;

– вектор-функция;

– функциональная матрица.

Задача аналитического конструирования агрегированных регуляторов (АКАР) формулируется следующим образом: требуется определить такой вектор управления , который обеспечивает перевод изображающей точки (ИТ) объекта (1.2.1) из произвольного исходного состояния (в некоторой допустимой области) сначала на многообразии , а затем движение вдоль них в заданное состояние (в частности, в начало координат x = 0). При этом движение ИТ обязательно должно удовлетворять системе основных функциональных уравнений метода АКАР:

(1.2.2)

Функции выбираются таким образом, чтобы, помимо асимптотической устойчивости (1.2.2), обеспечить желаемые показатели качества движения ИТ к притягивающим многообразиям . В простейшем случае . В общем случае, ИТ системы движется от одного многообразия к другому, пока не достигнет либо финишного многообразия, которое задаёт желаемый инвариант объекта (1.2.1), при скалярном управлений, либо финишных многообразий, которые задают желаемые инварианты объекта (1.2.1), при векторном управлении. Представленная выше задача АКАР может быть также сформулирована в терминах оптимального управления: функциональные уравнения (1.2.2) являются уравнениями Эйлера*-*Лагранжа для следующего обобщённого сопровождающего функционала (СОФ):

(1.2.3)

Таким образом, синтезированный методом АКАР вектор управления обеспечивает минимум СОФ, т.е. является оптимальным по СОФ. Функционал имеет второстепенную (сопровождающую) роль и в процедуре аналитического синтеза вектора управления непосредственно не участвует.

# **ГЛАВА 2. ПОСТАНОВКА ЗАДАЧИ УПРАВЛЕНИЯ**

## 2.1 Постановка задачи управления запасами однопродуктового слада

Рассмотрим простейшую модель управления запасами однопродуктового склада следующего вида:

,(2.1.1)

где количество товаров на складе в момент времени ;

количество товаров на складе в момент времени *t*;

объем поставок в момент времени *t*;

величина спроса в момент времени *t;*

параметр модели, который имеет фиксированное значение при моделировании и задан пользователем*;*

нормальный случайный процесс с нулевым средним и дисперсией равной *.*

Как видно из формулы (2.1.1) количество запасов в момент времени зависит от количества запасов в предыдущий момент времени, что говорит о наличии обратной связи в системе.

Для описания характера спроса в работе использовалась следующая модель:

,(2.1.2)

где величина спроса в момент времени

параметры модели, которые имеют фиксированные значения при моделировании и заданы пользователем

Уравнения (2.1.1) и (2.1.2) описывают динамику количества товаров на складе и спроса. Из уравнения (2.1.2) так же, как и из формулы (2.1.1) для определения количества товаров на складе, видна обратная связь в функции нахождения величины спроса на эти товары. Для описания стохастического характера количества товаров был введён случайный процесс.

Требования к управлению определялись с учётом грузоподъёмности транспортного средства, также для уменьшения транспортных затрат предполагается, что поставки осуществлялись только при снижении количества товаров на складе до значения:

(2.1.3)

где минимальное значение, при котором осуществлялись поставки, введённое для уменьшения затрат, связанных с транспортировкой товаров на склад;

оптимальный размер заказа осуществляемого в момент времени ;

максимальная грузоподъёмность транспортного средства необходимого для доставки заказа.

Таким образом, с учётом ограничения по формуле (2.1.3) получим правило управления запасами:

(2.1.4)

где — объём реальных поставок, полученных с учётом ограничений.

Предполагается, что начальное значение количества товаров на складе и начальное значение величины спроса *,* параметры модели а также ограничения на управление и задаются при инициализации модели.

Таким образом, конечное описание модели запасов для однопродуктового склада примет вид:

# **ГЛАВА 3. АНАЛИТИЧЕСКИЙ СИНТЕЗ РЕГУЛЯТОРОВ**

## 3.1 Синтез управления запасами на основе метода АКАР

Для применения стохастического аналога аналитического конструирования агрегированных регуляторов (АКАР) удобнее рассмотреть модель (2.1.1), (2.1.2), (2.1.4) в виде:

(3.1.1)

,

где *t* = 0, 1, 2 ….

Для объекта (3.1.1) ставится задача нахождения закона управления, обеспечивающего максимально желаемый уровень запасов на складе.

Задаётся целевое многообразие в виде:

,(3.1.2)

где — целевая макропеременная;

 — максимально заданный целевой уровень количества запасов на складе.

Требуется выполнение основного функционального разностного уравнения, согласно классическому детерминированному АКАР при фиксированных значениях :

(3.1.3)

На решениях уравнения (3.1.3) достигается минимум функционала (вариационная задача):

,(3.1.4)

где функционал качества синтезируемой системы управления;

параметр функционала, связь параметров и выражена соотношением:

Выводится закон управления из формулы (3.1.3) с учётом уравнения (3.1.1) и модели объекта (3.1.2):

(3.1.5)

Выразим u(t) из уравнения (3.1.5):

(3.1.6)

Получим структуру управления с шумом.

Вывод стохастического регулятора для объекта (3.1.1). Управление для объекта в условиях неизвестных (ограниченных шумов) найдём в виду условного математического ожидания (УМО):

где вектор случайной величины до момента времени .

После взятия условного математического ожидания.

(3.1.7)

Уточнённая структура управления с шумом

Далее необходимо исключить шум из уравнения (3.1.7) на основе декомпозиции исходного описания объекта (3.1.1), выполнив подстановку уравнения (3.1.7) в уравнение (3.1.1).

(3.1.8)

Из уравнения (3.1.2)

(3.1.9)

Подставим (3.1.8) в (3.1.9).

Выразим

Выразим

(3.1.10)

Подставим (3.1.10) в (3.1.7), тем самым получим итоговое выражение для управления запасами по методу АКАР стохастическом.

(3.1.11)

Таким образом, итоговое описание стохастической теории управления запасами составляет совокупность уравнений (3.1.1), (3.1.2), (3.1.11)

# **ГЛАВА 4. ОПИСАНИЕ РАЗРАБОТАННОЙ ПРОГРАММЫ**

## 4.1 Функциональные требования к программе

Разработанная программа, должна предоставлять конечному пользователю интерфейс для работы с математической моделью однопродуктового склада в условиях стохастической неопределённости. Программа должна обеспечивать выполнение следующих функций:

* ввод начальных параметров модели;
* расчёт модели для введённых начальных параметров;
* составление таблицы полученных результатов и сохранение её в отдельный файл;
* построение графиков и сохранение их в папку проекта;
* вывод уведомления для пользователя о об окончание выполнения вычисления программой.

## 4.2 Требования к используемому программному обеспечению

Для разработки программы была выбрана кроссплатформенная интегрированная среда разработки PyCharm для языка программирования Python, разработанная компанией IntelliJ IDEA. Язык Python был выбран из-за его простоты, удобства при обработки математических данных, большого количества библиотек для вычислений, построения графиков, создания графического интерфейса. Также имеется возможность лёгкой упаковки приложение в exe. файл, что позволяет быстро запустить разработанное приложение на любом компьютере для любой операционной системы, не имеющие среду разработки PyCharm.

Для создания графического интерфейса был выбран PyQt5 – набор библиотек для создания графического интерфейса на базе платформы Qt5 от компании Digia. Средой разработки графического интерфейса была выбрана платформа QtDesigner - кроссплатформенная свободная среда для разработки графических интерфейсов для программ, использующих библиотеку Qt. Данный выбор обоснован простотой использования, заключающейся в визуальном выстраивании объектов на форме в среде разработки. Разработанный интерфейс далее импортируется в программу, где далее прописывается функционал.

## 4.3 Алгоритм работы программы

Программа разделена на 5 вычислительных блока, каждый из которых выполняет свои функции.

В блоке 1 производится расчёт модели с ограничением на управление с изменяющейся для шума во внешнем цикле. Производится расчёт значений переменных . Для полученных значений вычисляются оценки математического ожидания, дисперсии, СКО, ОШС. Вычисляется энергия управления. Строятся графики для найденных величин.

В блоке 2 производится расчёт модели без ограничения на управление с изменяющейся для шума во внешнем цикле. Производится расчёт значений переменных . Для полученных значений вычисляются оценки математического ожидания, дисперсии, СКО, ОШС. Вычисляется энергия управления. Строятся графики для найденных величин.

В блоке 3 производится расчёт модели с ограничением на управление с постоянной . Производится расчёт значений переменных . Для полученных значений вычисляются оценки математического ожидания, дисперсии, СКО по ансамблю и выборке. Вычисляется энергия управления. Строятся графики для найденных величин. Строится таблица полученных данных для дальнейшего анализа и сохраняется в папке с проектом.

В блоке 4 производится расчёт модели с ограничением на управление с постоянной для анализа влияния на полученные результаты параметра регулятора. Производится расчёт значений переменных . Вычисляется энергия управления. Строятся графики для найденных величин.

В блоке 5 производится расчёт модели с ограничением на управление с постоянной для анализа влияния на полученные результаты начальных значений модели. Производится расчёт значений переменных . Вычисляется энергия управления. Строятся графики для найденных величин.

Все блоки программы работают по схожему алгоритмы, но представленные в них модели делятся на два типа: с ограничением на управление и без. Блок схемы для моделей представлены ниже на рисунке 1 и рисунке 2.

При запуске приложения функция, обрабатывающая нажатие кнопки пользователем, вызывает функцию для инициализации значений, введённых в форму. Производится расчёт модели для каждого блока, строятся графики и таблица. После чего пользователь видит всплывающее окно с информацией о завершении расчёта. Для дальней работы необходимо поменять интересующие пользователя значения и запустить расчёт снова.

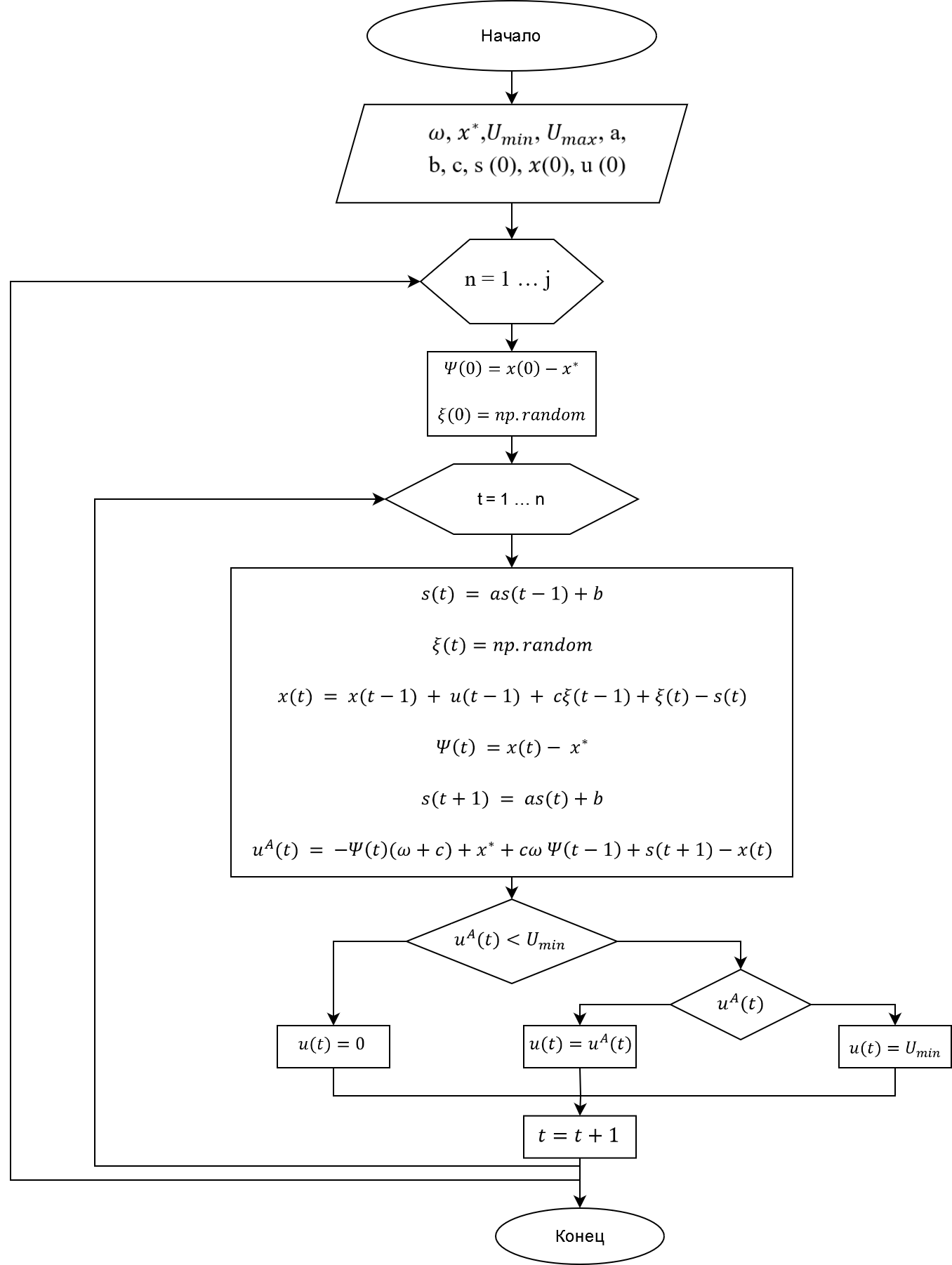


Рисунок 1 – Блок-схема для модели с ограничением на управление

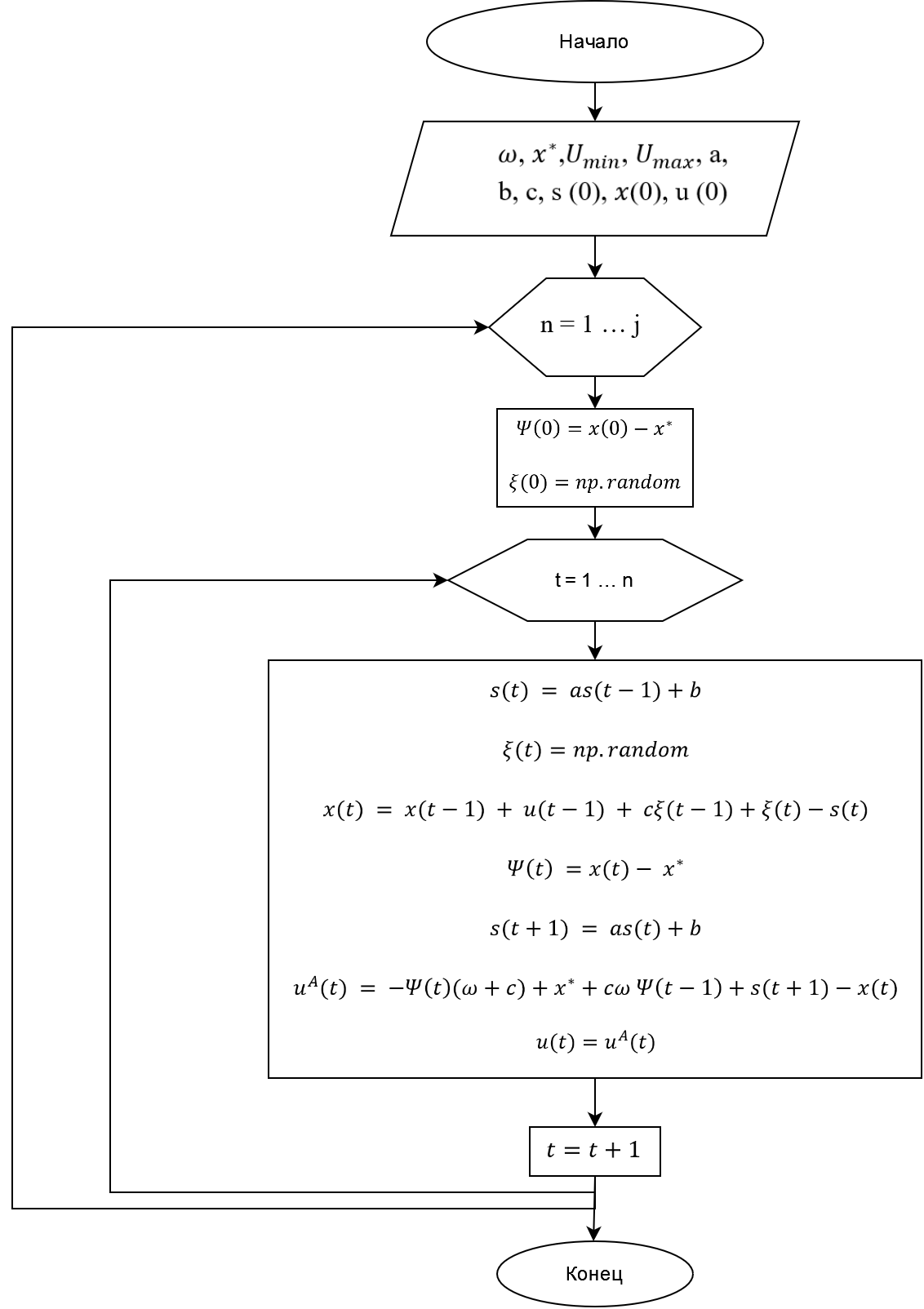


Рисунок 2 – Блок-схема для модели без ограничения на управление

## 4.4 Графический интерфейс

Разработанная форма для графического интерфейса представлена на рисунке 3.

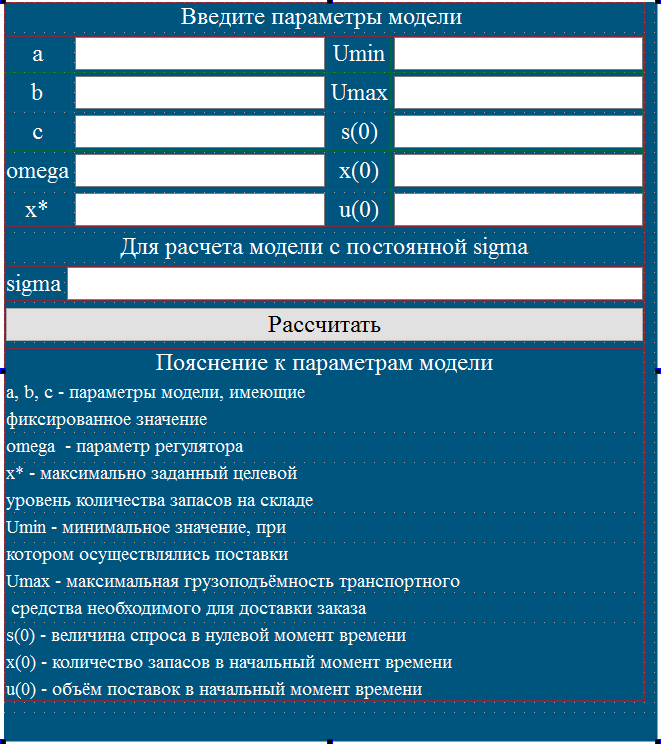


Рисунок 3 – mainwindow.ui

Описание к элементам формы представлено в таблице 1.

Таблица 1 – Описание элементов формы графического интерфейса

|  |  |
| --- | --- |
| Наименование виджета | Программное имя виджета |
| QLabel | for\_a\_2 |
| QLabel | for\_b\_2 |
| QLabel | label\_3 |
| QLabel | for\_c\_2 |
| QLabel | for\_w\_2 |
| QLabel | for\_x\_1 |
| QLabel | label\_10 |
| QLabel | label\_11 |
| QLabel | label\_7 |
| QLabel | label\_8 |
| QLabel | label\_9 |
| QLabel | label\_12 |
| QLabel | label\_13 |
| QLabel | label\_14 |
| QLineEdid | for\_Umax |
| QLineEdid | for\_Umin |
| QLineEdid | for\_a |
| QLineEdid | for\_b |
| QLineEdid | for\_c |
| QLineEdid | for\_s0 |
| QLineEdid | for\_u0 |
| QLineEdid | for\_w |
| QLineEdid | for\_x0 |
| QLineEdid | for\_x\_ |
| QLineEdid | for\_sigma |
| QPushButton | btn\_count |

# **ГЛАВА 5. ТЕСТИРОВАНИЕ**

## 5.1 Моделирование системы управления методом АКАР

Тестирование осуществлялось для модели с параметрами:

Результаты моделирования для модели запасов с ограничением на управление представлены на графике зависимости количества товара от времени на рисунке 4, на графике зависимости спроса от времени на рисунке 5, на графике зависимости величины поставок от времени на рисунке 6, на графике зависимости макропеременной от времени на рисунке 7. Параметры модели: .

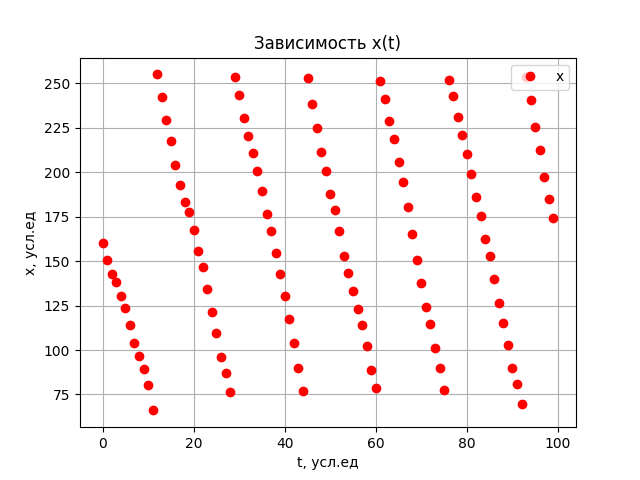


Рисунок 5 – График зависимости x(t),

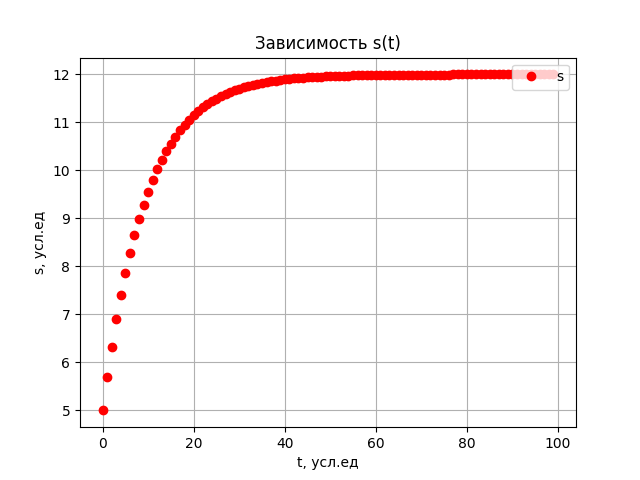


Рисунок 6 – График зависимости s(t),

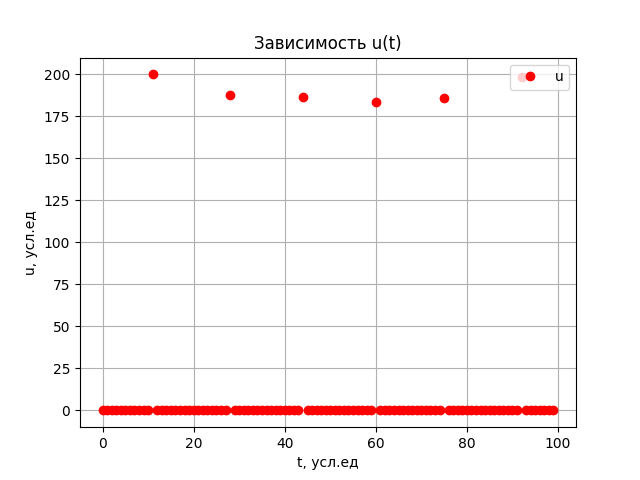


Рисунок 7 – График зависимости u(t),

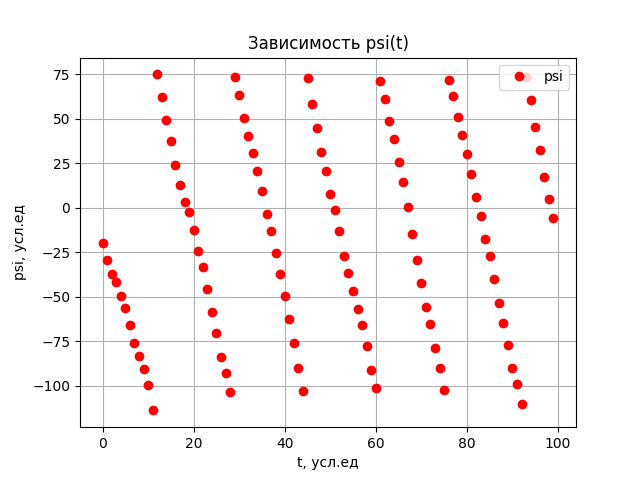


Рисунок 8 – График зависимости

Обсуждение результатов численного моделирования:

* в соответствии с рисунком 6 спрос постепенно увеличивался и уравновесился в районе 12 условных единиц;
* в соответствии с рисунком 7 в течение 100 условных временных промежутков было произведено 6 поставок, величина которых лежит в промежутке [175; 200];
* в соответствии с рисунком 5 количество товаров постепенно уменьшалось, достигнув минимума пополнялось поставками которых было 6, что совпадает с результатами полученными из рисунка 7.

Результаты моделирования для модели запасов с ограничением на управление представлены на графике зависимости количества товара от времени на рисунке 9, на графике зависимости спроса от времени на рисунке 10, на графике зависимости величины поставок от времени на рисунке 11, на графике зависимости макропеременной от времени на рисунке 12. Параметры модели: .

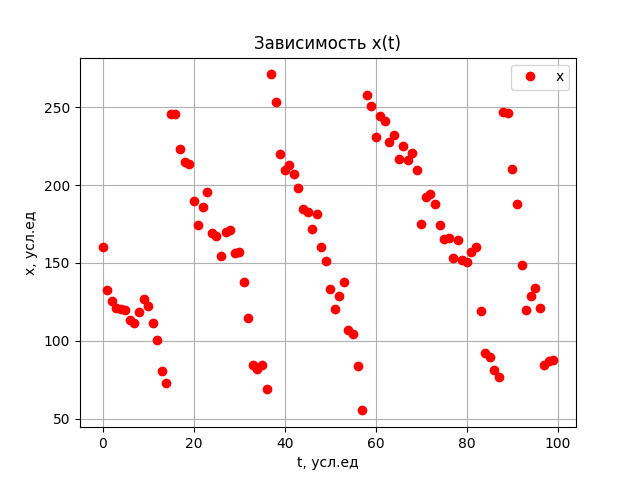


Рисунок 9 – График зависимости x(t),

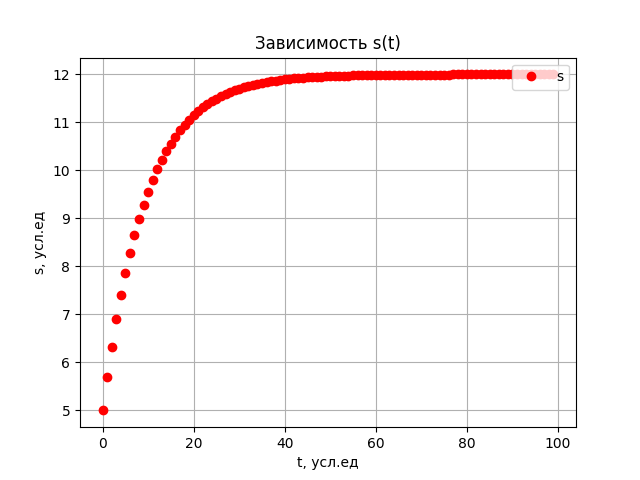


Рисунок 10 – График зависимости s(t),

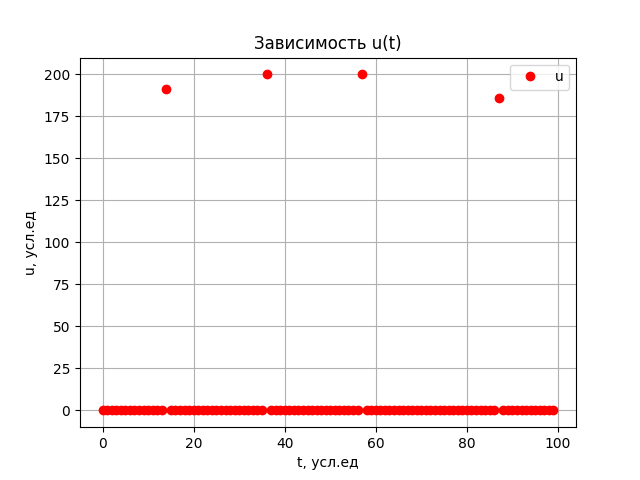


Рисунок 11 – График зависимости u(t),

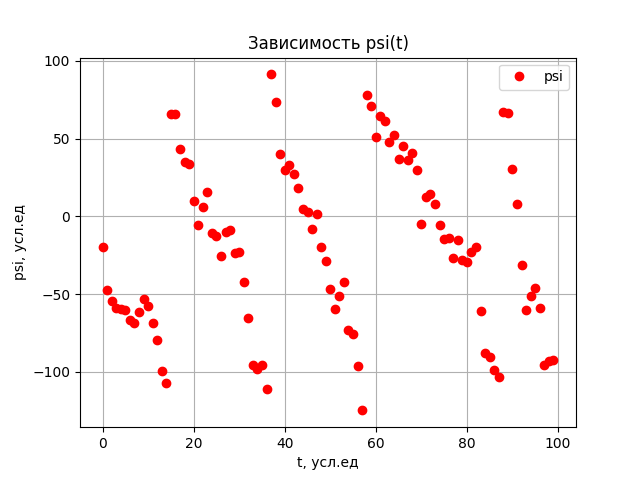


Рисунок 12 – График зависимости ,

Обсуждение результатов численного моделирования:

* в соответствии с рисунком 10 спрос постепенно увеличивался и уравновесился в районе 12 условных единиц;
* в соответствии с рисунком 11 в течение 100 условных временных промежутков было произведено 4 поставки, величина которых лежит в промежутке [175; 200];
* в соответствии с рисунком 9 количество товаров постепенно уменьшалось, достигнув минимума пополнялось поставками которых было 4, что совпадает с результатами полученными из рисунка 11. За счёт увеличения значения с 1.6 до 12.8 для получения шума на графике кол-ва товаров пропали чёткие очертания линий, что объяснимо колебанием значений случайной величины.

Результаты моделирования для модели запасов с ограничением на управление представлены на графике зависимости количества товара от времени на рисунке 13, на графике зависимости спроса от времени на рисунке 14, на графике зависимости величины поставок от времени на рисунке 15, на графике зависимости макропеременной от времени на рисунке 16. Параметры модели: .

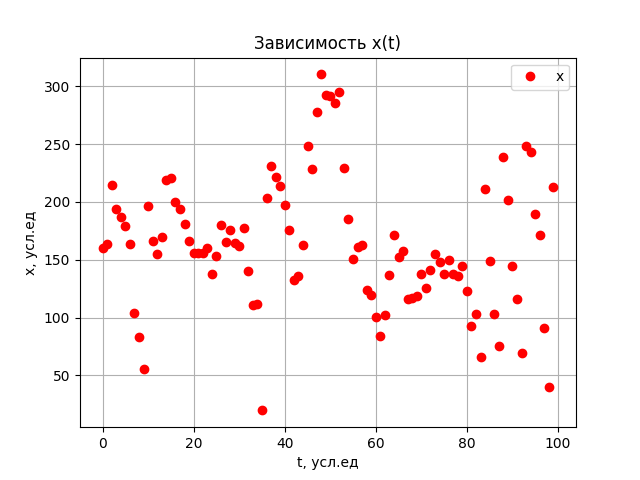


Рисунок 13 – График зависимости x(t),

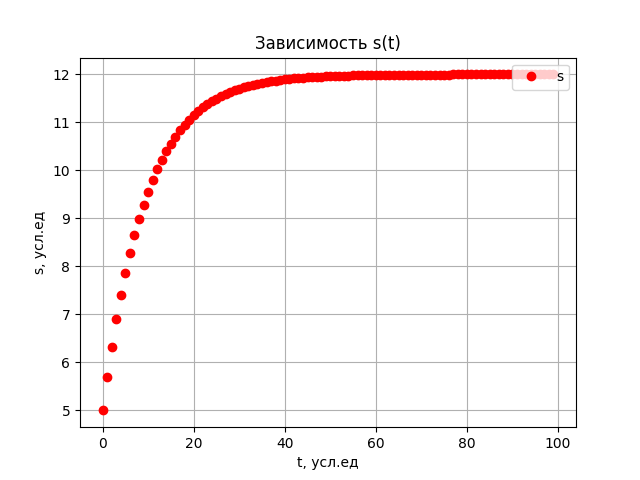


Рисунок 14 – График зависимости s(t),

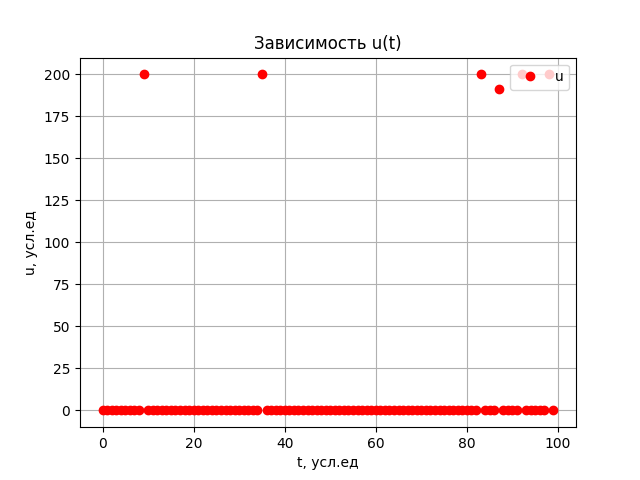


Рисунок 15– График зависимости u(t),

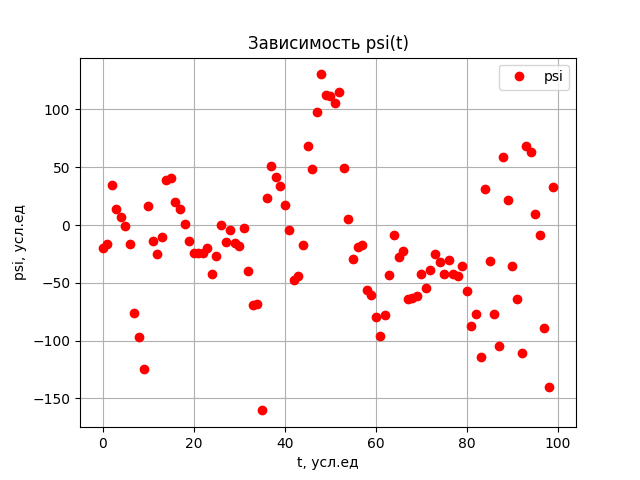


Рисунок 16 – График зависимости

Обсуждение результатов численного моделирования:

* в соответствии с рисунком 14 спрос постепенно увеличивался и уравновесился в районе 12 условных единиц;
* в соответствии с рисунком 15 в течение 100 условных временных промежутков было произведено 6 поставок, величина которых лежит в промежутке [175; 200];
* в соответствии с рисунком 13 количество товаров постепенно уменьшалось, достигнув минимума пополнялось поставками которых было 6, что совпадает с результатами полученными из рисунка 15. За счёт увеличения значения с 12.8 до 24 для получения шума на графике кол-ва товаров пропали чёткие очертания линий, что объяснимо колебанием значений случайной величины, однако пополнение склада все же проследит можно.

Результаты моделирования для моделей запасов с ограничением на управления и без представлены на графике зависимости количества товара от времени на рисунке 17, на графике зависимости спроса от времени на рисунке 18, на графике зависимости величины поставок от времени на рисунке 19, на графике зависимости макропеременной от времени на рисунке 20. Параметры модели: .

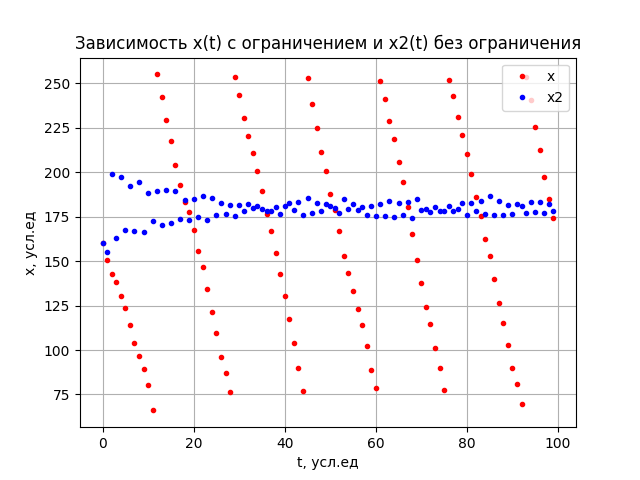


Рисунок 17 – График зависимости x(t),

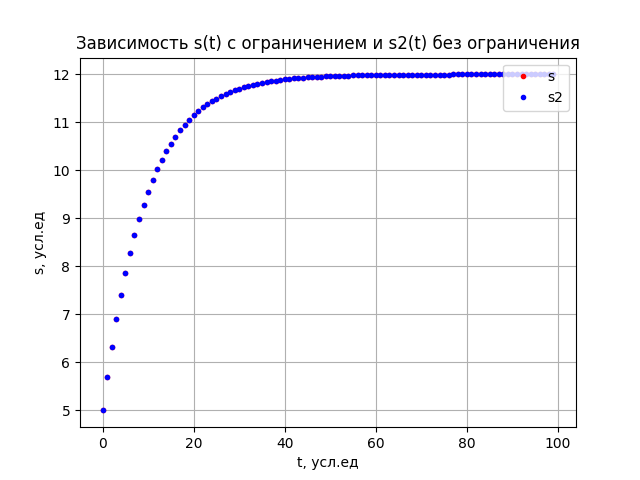


Рисунок 18 – График зависимости s(t),

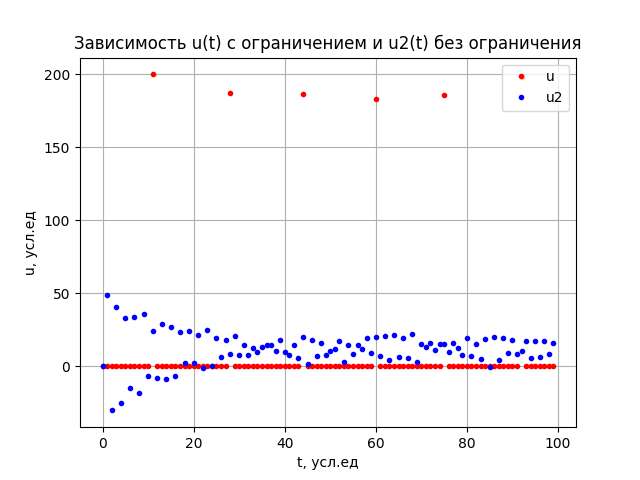


Рисунок 19 – График зависимости u(t),

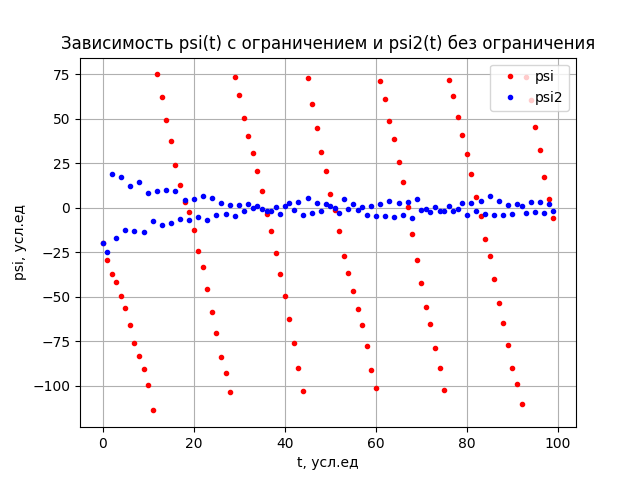


Рисунок 20 – График зависимости

Обсуждение результатов численного моделирования:

* в соответствии с рисунком 18 спрос постепенно увеличивался и уравновесился в районе 12 условных единиц для обоих вариантов модели;
* в соответствии с рисунком 19 в течение 100 условных временных промежутков было произведено 6 поставок для модели с ограничением на управление, величина которых лежит в промежутке [175; 200]. Однако, для модели без ограничений поставки производятся практически в каждый условный момент времени, что говорит о дополнительных растратах фирмы/производства на транспортировку товара. Объём поставок колеблется в промежутке [-50; 50];
* в соответствии с рисунком 17 для модели с ограничением количество товаров постепенно уменьшалось, достигнув минимума, пополнялось поставками, которых было 6, что совпадает с результатами полученными из рисунка 19. В случае же без ограничений, склад опустошался и пополнялся неравномерно.

Результаты моделирования для моделей запасов с ограничением на управления и без представлены на графике зависимости количества товара от времени на рисунке 21, на графике зависимости спроса от времени на рисунке 22, на графике зависимости величины поставок от времени на рисунке 23, на графике зависимости макропеременной от времени на рисунке 24. Параметры модели: .

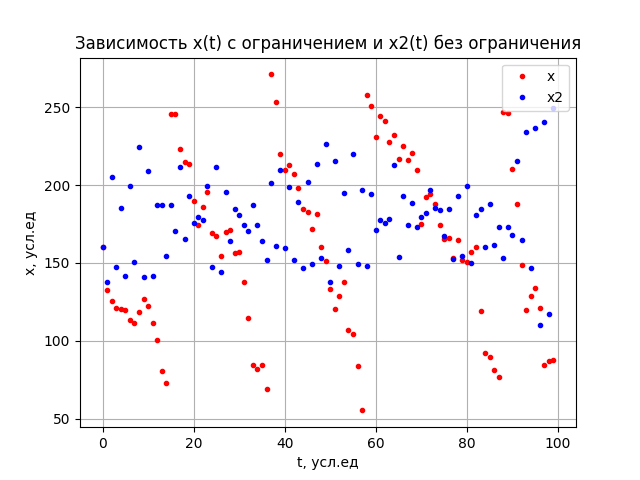


Рисунок 21 – График зависимости x(t)

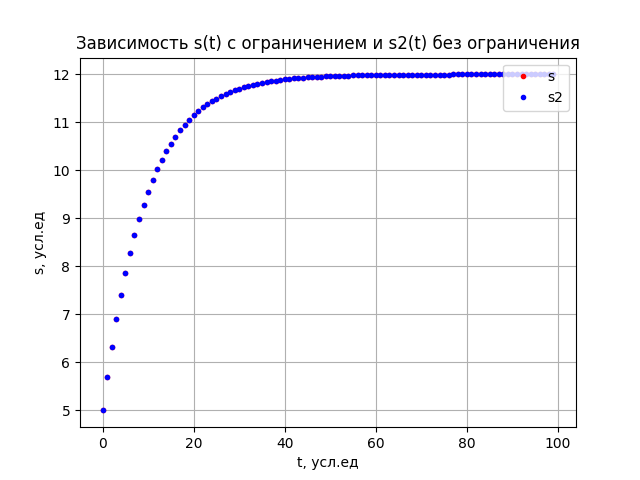


Рисунок 22 – График зависимости s(t),

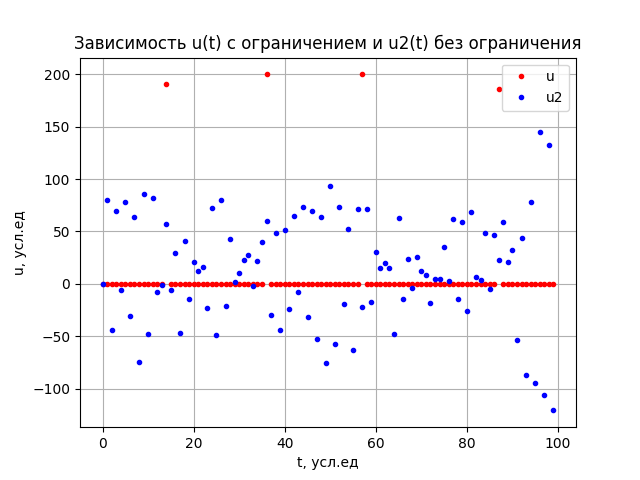


Рисунок 23 – График зависимости u(t),

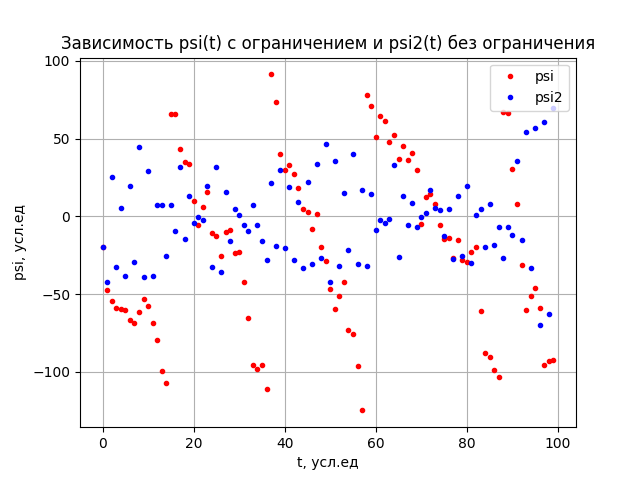


Рисунок 24 – График зависимости

Обсуждение результатов численного моделирования:

* в соответствии с рисунком 22 спрос постепенно увеличивался и уравновесился в районе 12 условных единиц для обоих вариантов модели;
* в соответствии с рисунком 23 в течение 100 условных временных промежутков было произведено 4 поставки для модели с ограничением на управление, величина которых лежит в промежутке [175; 200]. Однако, для модели без ограничений поставки производятся практически в каждый условный момент времени, что говорит о дополнительных растратах фирмы/производства на транспортировку товара. Объём поставок колеблется в промежутке от [-150; 150];
* в соответствии с рисунком 21 для модели с ограничением количество товаров постепенно уменьшалось, достигнув минимума, пополнялось поставками, которых было 4, что совпадает с результатами полученными из рисунка 23. В случае же без ограничений, склад опустошался и пополнялся неравномерно. За счёт увеличения значения с 1.6 до 12.8 для получения шума на графике кол-ва товаров пропали чёткие очертания линий, что объяснимо колебанием значений случайной величины, однако пополнение склада все же проследит можно.

Результаты моделирования для моделей запасов с ограничением на управления и без представлены на графике зависимости количества товара от времени на рисунке 25, на графике зависимости спроса от времени на рисунке 26, на графике зависимости величины поставок от времени на рисунке 27, на графике зависимости макропеременной от времени на рисунке 28. Параметры модели: .

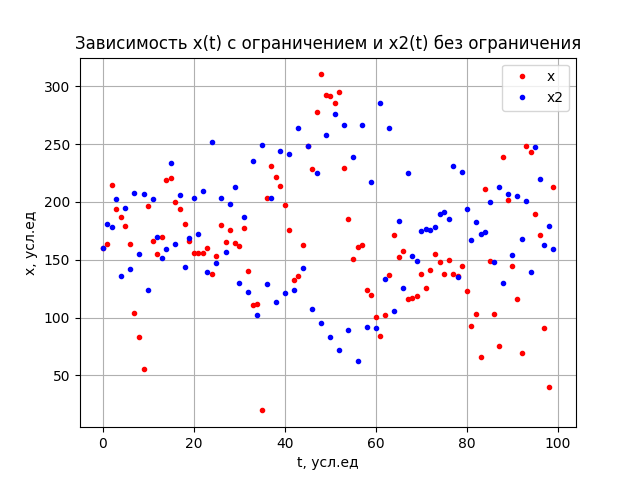


Рисунок 25 – График зависимости x(t),

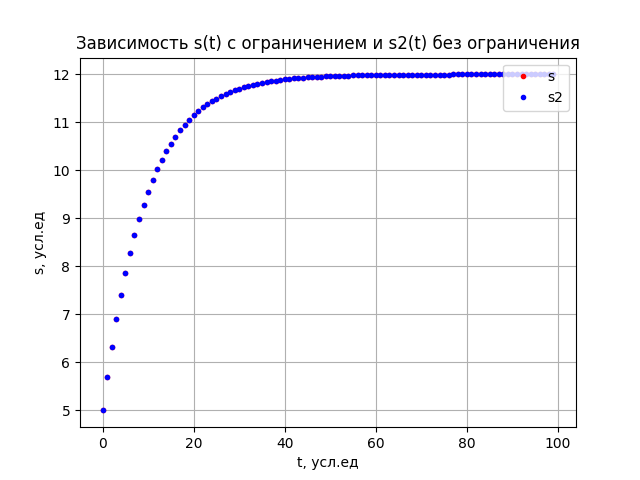


Рисунок 26 – График зависимости s(t),

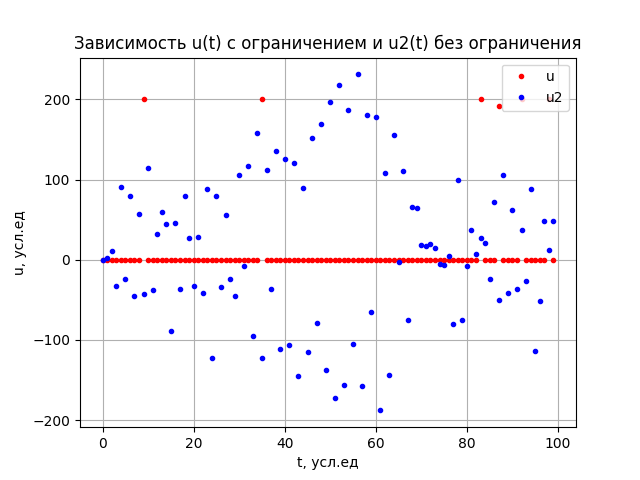


Рисунок 27 – График зависимости u(t),

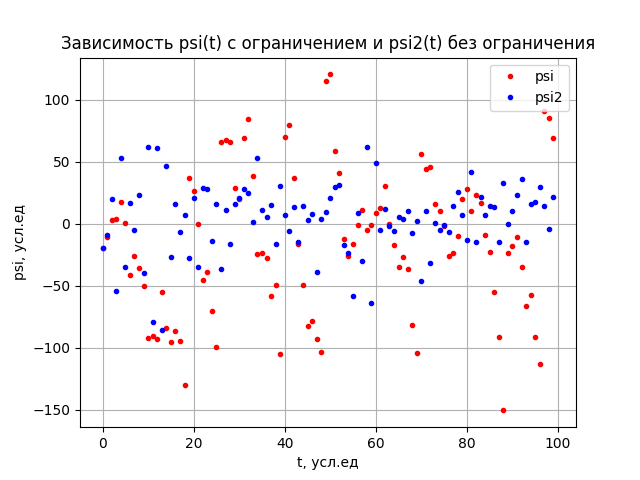


Рисунок 28 – График зависимости ,

Обсуждение результатов численного моделирования:

* в соответствии с рисунком 26 спрос постепенно увеличивался и уравновесился в районе 12 условных единиц для обоих вариантов модели;
* в соответствии с рисунком 27 в течение 100 условных временных промежутков было произведено 5 поставок для модели с ограничением на управление, величина которых лежит в промежутке [175; 200]. Однако, для модели без ограничений поставки производятся практически в каждый условный момент времени, что говорит о дополнительных растратах фирмы/производства на транспортировку товара. Объём поставок колеблется в промежутке от [-150; 200];
* в соответствии с рисунком 25 для модели с ограничением количество товаров постепенно уменьшалось, достигнув минимума, пополнялось поставками, которых было 5, что совпадает с результатами полученными из рисунка 27. В случае же без ограничений, склад опустошался и пополнялся неравномерно. За счёт увеличения значения с 12.8 до 24 для получения шума на графике кол-ва товаров пропали чёткие очертания линий, что объяснимо колебанием значений случайной величины, однако пополнение склада все же проследит можно.

Результаты моделирования для модели запасов с ограничением на управления и без представлены на графиках зависимости количества товара от времени на рисунке 29 и рисунке 30, на графиках зависимости спроса от времени на рисунке 31 и рисунке 32, на графиках зависимости величины поставок от времени на рисунке 33 и на рисунке 34, на графике зависимости макропеременной от времени на рисунках 35 и 36. Для эксперимента были взяты результаты для одной и той же модели в 1 и 50 прогон для сравнения полученных результатов. Параметры модели: .

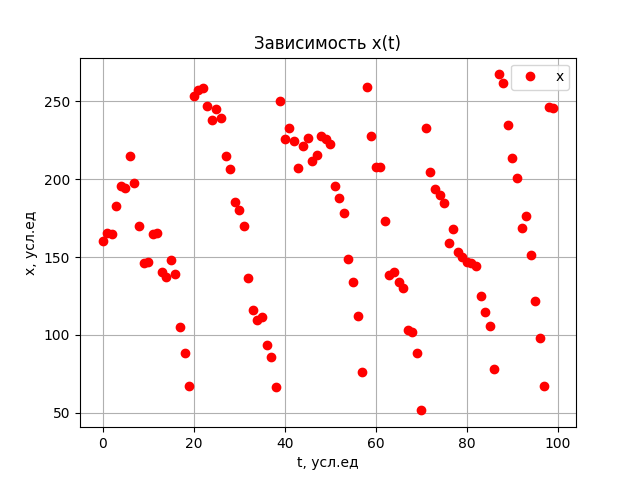


Рисунок 29 – График зависимости x(t) в 1 прогон,

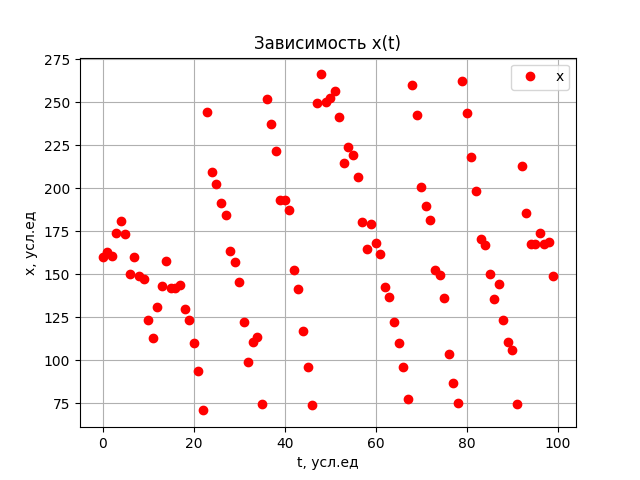


Рисунок 30 – График зависимости x(t) в 50 прогон,

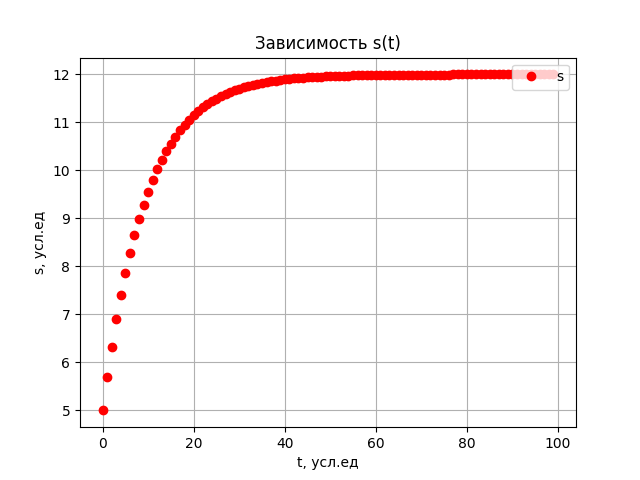


Рисунок 31 – График зависимости s(t) в 1 прогон,

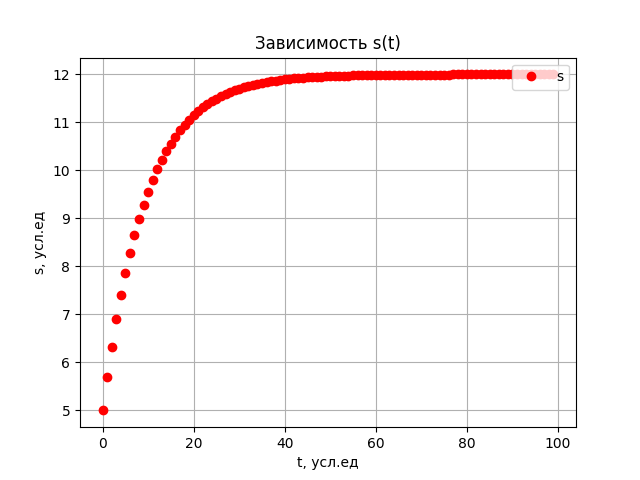


Рисунок 32 – График зависимости s(t) в 50 прогон,

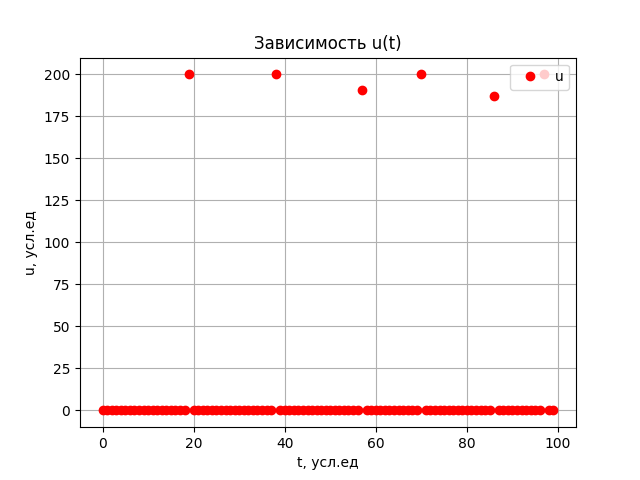


Рисунок 33 – График зависимости u(t) в 1 прогон,

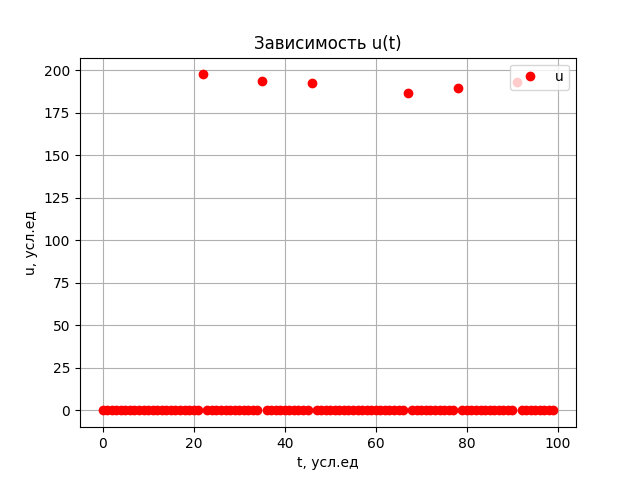


Рисунок 34 – График зависимости u(t) в 50 прогон,

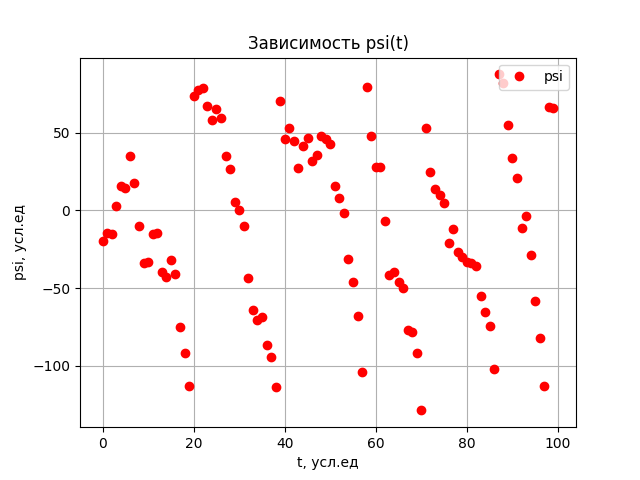


Рисунок 35 – График зависимости в 1 прогон,

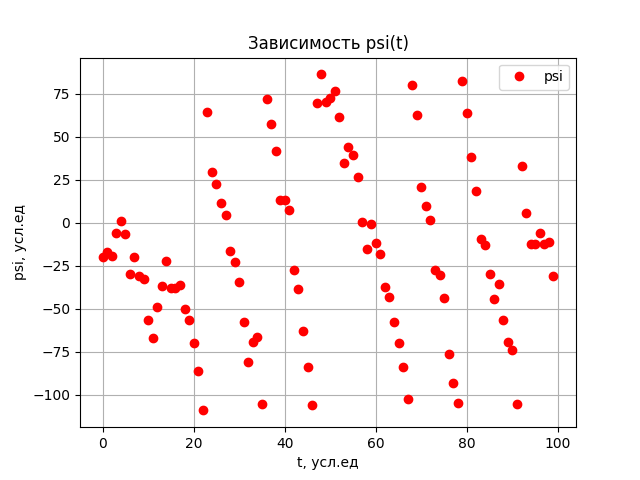


Рисунок 36 – График зависимости в 50 прогон,

Обсуждением результатов численного моделирования: значения, полученные для исследуемых переменных схожи, но имеют незначительные различия, что объяснимо стохастической природой модели запасов.

## 5.2 Оценка качества полученных моделей

В блоке 1 программы для математической модели с ограничением на управление с изменяющейся были найдены оценки математического ожидания, дисперсии, среднего квадратического отклонения для каждой из исследуемых переменных при каждом значении из списка по формулам, представленным ниже. Параметры модели:

,

Оценка математического ожидания:

, (5.2.1)

где оценка математического ожидания количества товаров в момент времени t;

количество реализаций модели;

значение количества товаров в *i-*й момент времени *t*.

Оценка дисперсии:

, (5.2.2)

где оценка дисперсии количества товаров в момент времени t.

Оценка среднего квадратического отклонения:

, (5.2.3)

где оценка среднего квадратического отклонения количества товаров в момент времени t.

Для построения графиков зависимости найденных оценок от отношения шум/сигнал (ОШС) были просчитаны значения ОШС для каждой из исследуемых переменных при каждом значения из списка по формуле:

, (5.2.4)

где значение отношения шум/сигнал (ОШС);

константа модели;

среднее квадратическое отклонение для шума, введённое пользователем.

Значение энергии вычислялось по формуле:

,(5.2.5)

где количество затраченной энергии на обеспечение управления;

количество временных тактов;

энергия, затраченная в момент времени *i.*

Графики зависимости оценкок математического ожидания от ОШС для каждой из исследуемых переменных модели, описанной выше, представлены на рисунке 37 для количества товара, на рисунке 38 для спроса, на рисунке 39 для величины поставок, на рисунке 40 для макропеременной. Значения математических ожиданий и ОШС приведены в таблице 2 для количества товаров, в таблице 3 для спроса, в таблице 4 для величины поставок, в таблице 5 для макропеременной.

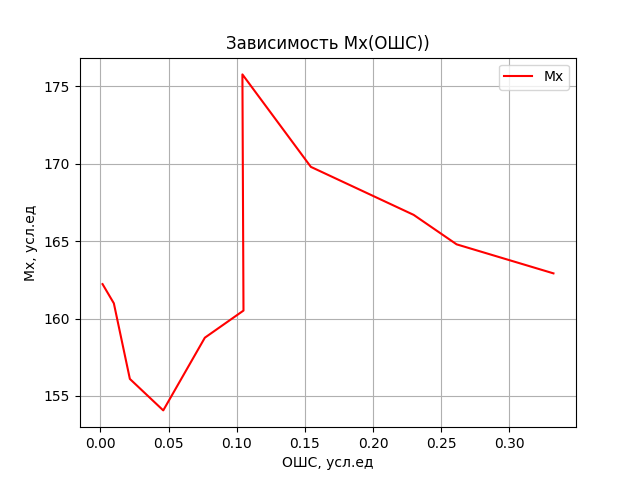


Рисунок 37 – График зависимости от ОШС

Таблица 2 – Значения переменных для рисунка 37

|  |  |  |
| --- | --- | --- |
|  |  | ОШС |
| 1.6 | 162.22 | 0.0016 |
| 3.84 | 160.98 | 0.0099 |
| 6.08 | 156.10 | 0.0216 |
| 8.32 | 154.06 | 0.0461 |
| 10.56 | 158.76 | 0.0766 |
| 12.8 | 160.51 | 0.1049 |
| 15.04 | 175.77 | 0.1041 |
| 17.28 | 169.80 | 0.1544 |
| 19.52 | 166.70 | 0.2298 |
| 21.76 | 164.79 | 0.2613 |
| 24 | 162.92 | 0.3323 |

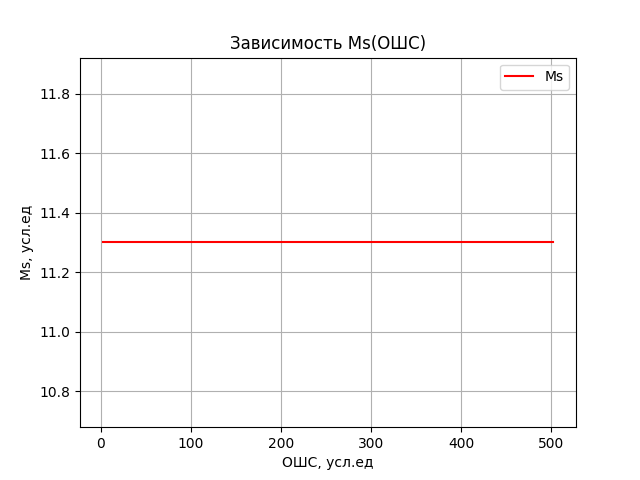


Рисунок 38 – График зависимости от ОШС

Таблица 3 – Значения переменных для рисунка 38

|  |  |  |
| --- | --- | --- |
|  |  | ОШС |
| 1.6 | 11.30 | 2.2334 |
| 3.84 | 11.30 | 12.8646 |
| 6.08 | 11.30 | 32.2508 |
| 8.32 | 11.30 | 60.3922 |
| 10.56 | 11.30 | 97.2887 |
| 12.8 | 11.30 | 142.9402 |
| 15.04 | 11.30 | 197.3468 |
| 17.28 | 11.30 | 260.5086 |
| 19.52 | 11.30 | 332.4254 |
| 21.76 | 11.30 | 413.0973 |
| 24 | 11.30 | 502.5243 |

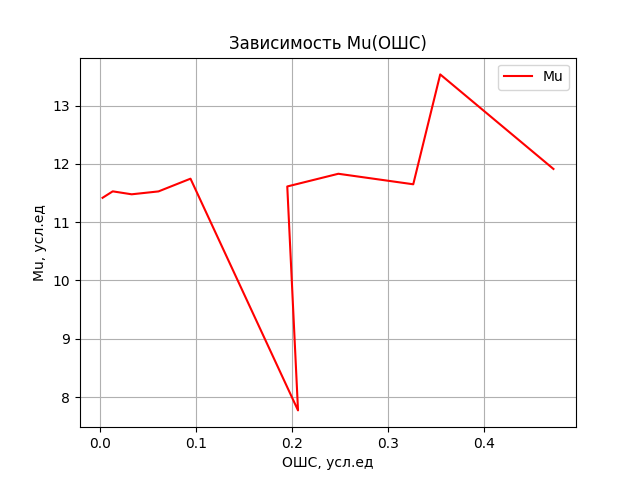


Рисунок 39 – График зависимости от ОШС

Таблица 4 – Значения переменных для рисунка 39

|  |  |  |
| --- | --- | --- |
|  |  | ОШС |
| 1.6 | 11.41 | 0.0022 |
| 3.84 | 11.53 | 0.0128 |
| 6.08 | 11.47 | 0.0326 |
| 8.32 | 11.52 | 0.0605 |
| 10.56 | 11.74 | 0.0938 |
| 12.8 | 7.77 | 0.2058 |
| 15.04 | 11.61 | 0.1946 |
| 17.28 | 11.83 | 0.2478 |
| 19.52 | 11.65 | 0.3258 |
| 21.76 | 13.53 | 0.3539 |
| 24 | 11.91 | 0.4718 |

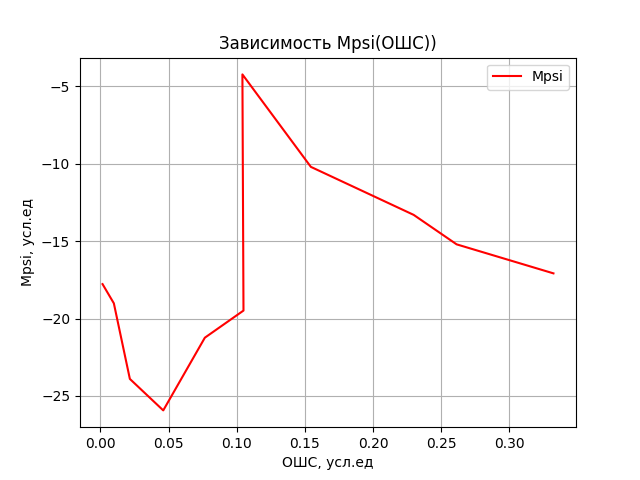


Рисунок 40 – График зависимости от ОШС

Таблица 5– Значения переменных для рисунка 40

|  |  |  |
| --- | --- | --- |
|  |  | ОШС |
| 1.6 | -17.77 | 0.0015 |
| 3.84 | -19.01 | 0.0098 |
| 6.08 | -23.90 | 0.0216 |
| 8.32 | -25.93 | 0.0460 |
| 10.56 | -21.23 | 0.0766 |
| 12.8 | -19.48 | 0.1049 |
| 15.04 | -4.22 | 0.1041 |
| 17.28 | -10.20 | 0.1543 |
| 19.52 | -13.30 | 0.2297 |
| 21.76 | -15.20 | 0.2612 |
| 24 | -17.08 | 0.3323 |

Графики зависимости оценкок математического ожидания от ОШС c учетом СКО для каждой из исследуемых переменных модели, описанной выше, представлены на рисунке 41 для количества товара, на рисунке 42 для спроса, на рисунке 43 для величины поставок, на рисунке 44 для макропеременной.

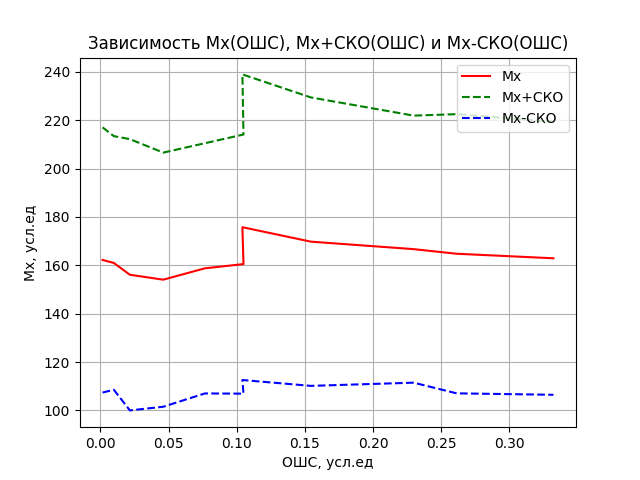


Рисунок 41 – График зависимости с учётом СКО от ОШС

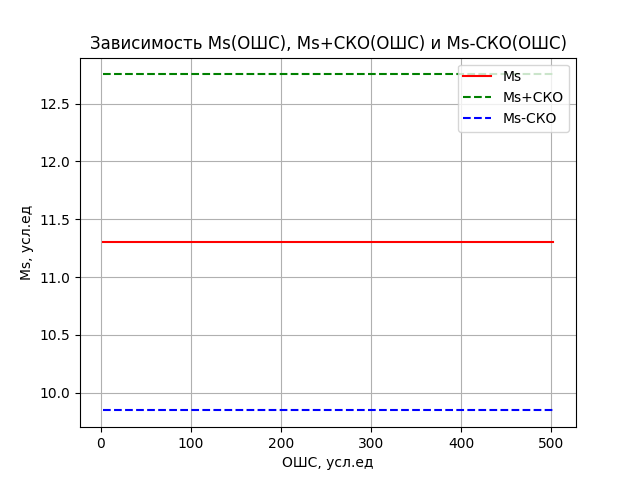


Рисунок 42 – График зависимости с учётом СКО от ОШС

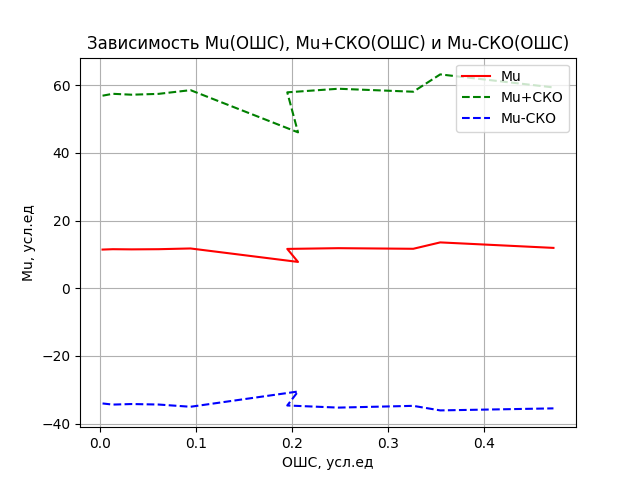


Рисунок 43 – График зависимости с учётом СКО от ОШС

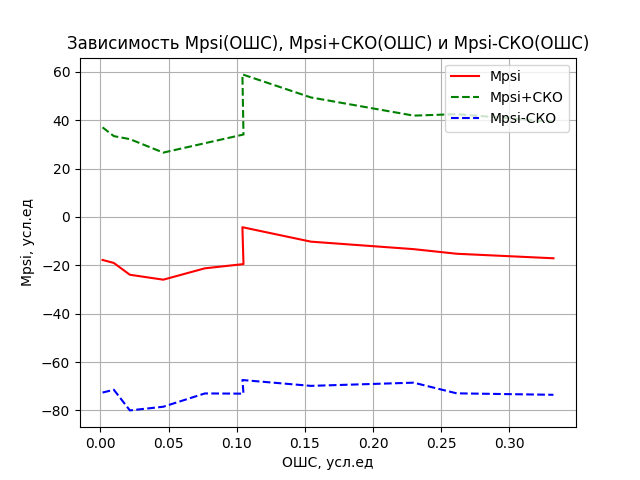


Рисунок 44 – График зависимости с учётом СКО от ОШС

На рисунке 45 представлен график зависимости количества затраченной энергии на обеспечение управления от ОШС для математической модели, описанной выше. Значения для количества энергии и ОШС для переменных для каждого значения представлены в таблице 6.

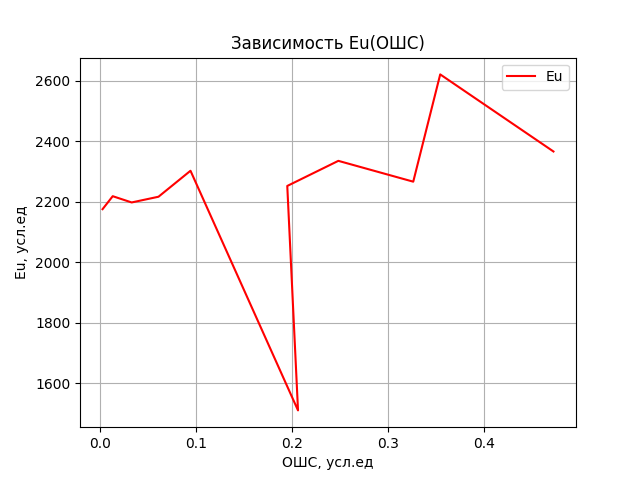


Рисунок 45 – График зависимости от ОШС

Таблица 6 – Значения переменных для рисунка 45

|  |  |  |
| --- | --- | --- |
|  |  | ОШС |
| 1.6 | 2175.85 | 0.0022 |
| 3.84 | 2218.77 | 0.0128 |
| 6.08 | 2198.10 | 0.0326 |
| 8.32 | 2216.98 | 0.0605 |
| 10.56 | 2302.85 | 0.0938 |
| 12.8 | 1510.86 | 0.2058 |
| 15.04 | 2252.64 | 0.1946 |
| 17.28 | 2335.59 | 0.2478 |
| 19.52 | 2266.76 | 0.3258 |
| 21.76 | 2621.29 | 0.3539 |
| 24 | 2366.53 | 0.4718 |

В блоке 2 программы для математической модели без ограничения на управление с изменяющейся были найдены оценки математического ожидания, дисперсии, среднего квадратического отклонения для каждой из исследуемых переменных при каждом значении из списка по формулам, представленным выше. Параметры модели:

,

.

Графики зависимости оценкок математического ожидания от ОШС для каждой из исследуемых переменных модели, описанной выше, представлены на рисунке 46 для количества товара, на рисунке 47 для спроса, на рисунке 48 для величины поставок, на рисунке 49 для макропеременной. Значения математических ожиданий и ОШС приведены в таблице 7 для количества товаров, в таблице 8 для спроса, в таблице 9 для величины поставок, в таблице 10 для макропеременной.

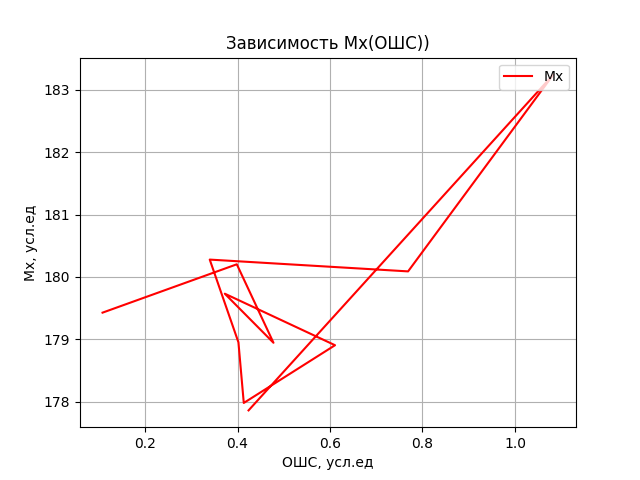


Рисунок 46 – График зависимости от ОШС

Таблица 7 – Значения переменных для рисунка 46

|  |  |  |
| --- | --- | --- |
|  |  | ОШС |
| 1.6 | 179.42 | 0.1081 |
| 3.84 | 180.20 | 0.3984 |
| 6.08 | 178.94 | 0.4778 |
| 8.32 | 179.73 | 0.3724 |
| 10.56 | 178.90 | 0.6108 |
| 12.8 | 177.97 | 0.4135 |
| 15.04 | 178.95 | 0.4019 |
| 17.28 | 180.27 | 0.3399 |
| 19.52 | 180.08 | 0.7688 |
| 21.76 | 183.24 | 1.0832 |
| 24 | 177.86 | 0.4238 |

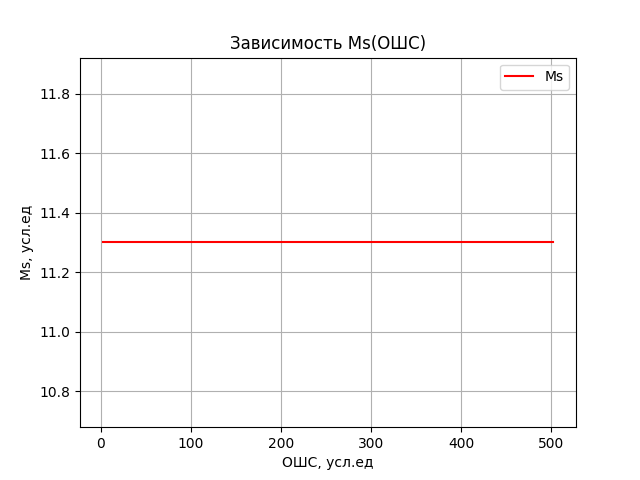


Рисунок 47 – График зависимости от ОШС

Таблица 8 – Значения переменных для рисунка 47

|  |  |  |
| --- | --- | --- |
|  |  | ОШС |
| 1.6 | 11.30 | 2.2334 |
| 3.84 | 11.30 | 12.8646 |
| 6.08 | 11.30 | 32.2508 |
| 8.32 | 11.30 | 60.3922 |
| 10.56 | 11.30 | 97.2887 |
| 12.8 | 11.30 | 142.9402 |
| 15.04 | 11.30 | 197.3468 |
| 17.28 | 11.30 | 260.5086 |
| 19.52 | 11.30 | 332.4254 |
| 21.76 | 11.30 | 413.0973 |
| 24 | 11.30 | 502.5243 |

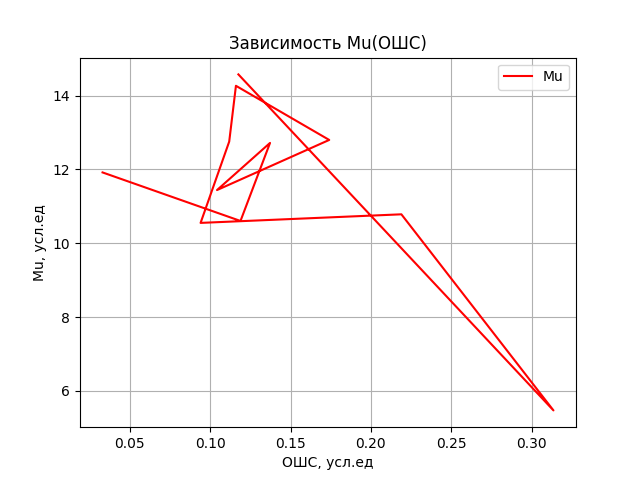


Рисунок 48 – График зависимости от ОШС

Таблица 9 – Значения переменных для рисунка 48

|  |  |  |
| --- | --- | --- |
|  |  | ОШС |
| 1.6 | 11.91 | 0.0329 |
| 3.84 | 10.60 | 0.1188 |
| 6.08 | 12.71 | 0.1372 |
| 8.32 | 11.44 | 0.1042 |
| 10.56 | 12.79 | 0.1740 |
| 12.8 | 14.26 | 0.1159 |
| 15.04 | 12.75 | 0.1118 |
| 17.28 | 10.55 | 0.0939 |
| 19.52 | 10.78 | 0.2189 |
| 21.76 | 5.47 | 0.3136 |
| 24 | 14.57 | 0.1175 |

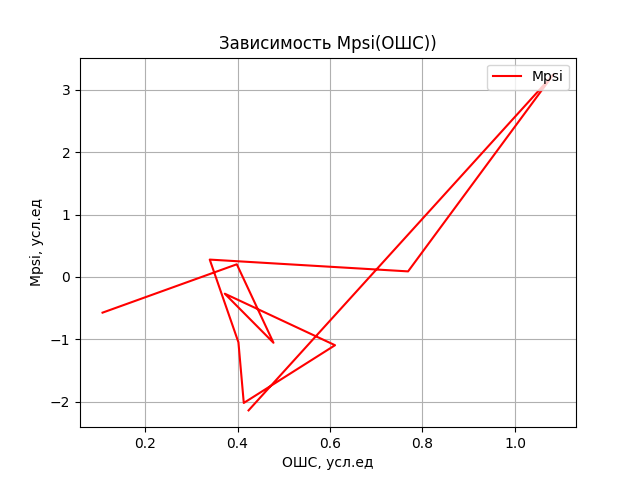


Рисунок 49 – График зависимости от ОШС

Таблица 10 – Значения переменных для рисунка 49

|  |  |  |
| --- | --- | --- |
|  |  | ОШС |
| 1.6 | -0.57 | 0.1081 |
| 3.84 | 0.20 | 0.3984 |
| 6.08 | -1.05 | 0.4778 |
| 8.32 | -0.26 | 0.3724 |
| 10.56 | -1.09 | 0.6108 |
| 12.8 | -2.02 | 0.4135 |
| 15.04 | -1.04 | 0.4019 |
| 17.28 | 0.27 | 0.3399 |
| 19.52 | 0.08 | 0.7688 |
| 21.76 | 3.24 | 1.0832 |
| 24 | -2.13 | 0.4238 |

Графики зависимости оценкок математического ожидания от ОШС c учетом СКО для каждой из исследуемых переменных модели, описанной выше, представлены на рисунке 50 для количества товара, на рисунке 51 для спроса, на рисунке 52 для величины поставок, на рисунке 53 для макропеременной.

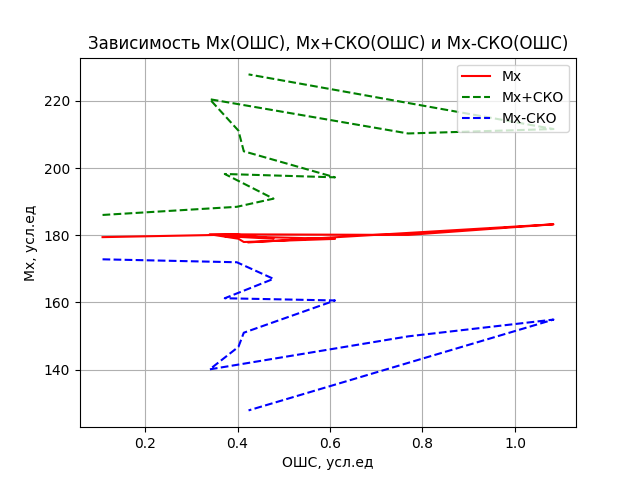


Рисунок 50 – График зависимости с учётом СКО от ОШС

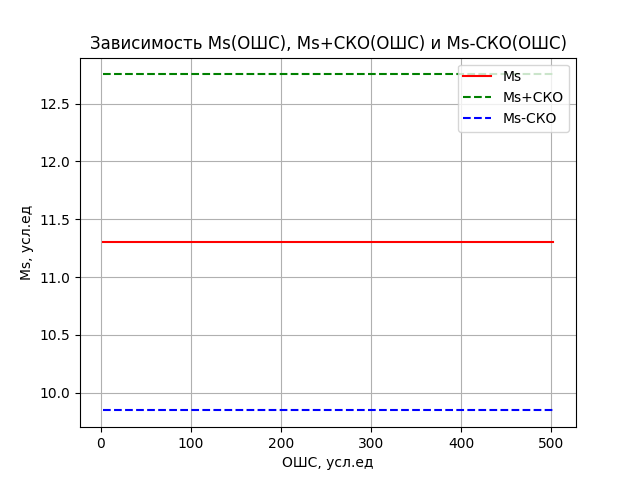


Рисунок 51 – График зависимости с учётом СКО от ОШС

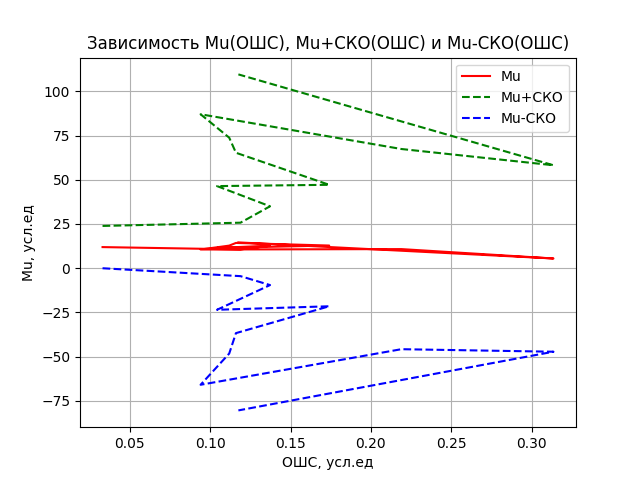


Рисунок 52 – График зависимости с учётом СКО от ОШС

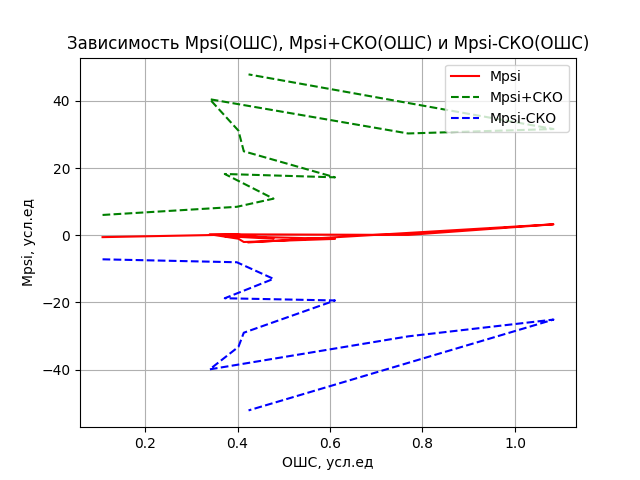


Рисунок 53 – График зависимости с учётом СКО от ОШС

На рисунке 54 представлен график зависимости количества затраченной энергии на обеспечение управления от ОШС для математической модели, описанной выше. Значения энергии и ОШС приведены в таблице 11.

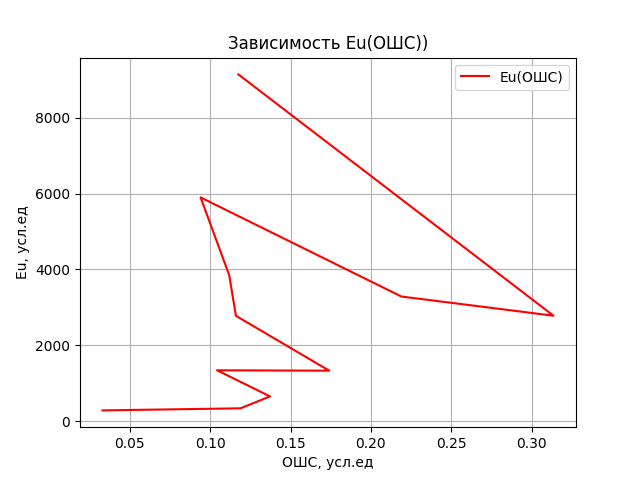


Рисунок 54 – График зависимости от ОШС

Таблица 11 – Значения переменных для рисунка 54

|  |  |  |
| --- | --- | --- |
|  |  | ОШС |
| 1.6 | 283.62 | 0.0329 |
| 3.84 | 338.59 | 0.1188 |
| 6.08 | 652.68 | 0.1372 |
| 8.32 | 1341.45 | 0.1042 |
| 10.56 | 1331.39 | 0.1740 |
| 12.8 | 2777.71 | 0.1159 |
| 15.04 | 3849.86 | 0.1118 |
| 17.28 | 5901.59 | 0.0939 |
| 19.52 | 3287.29 | 0.2189 |
| 21.76 | 2781.16 | 0.3136 |
| 24 | 9145.33 | 0.1175 |

В блоке 3 программы для математической модели с ограничением на управление были найдены оценки математического ожидания, дисперсии, среднего квадратического отклонения по 50 ансамблям и 100 временным промежуткам для каждой из исследуемых переменных по формулам, представленным выше. Параметры модели: ,

.

Графики оценкок математического ожидания с учетом СКО по 100 временным промежуткам для каждой из исследуемых переменных модели, описанной выше, представлены на рисунке 55 для количества товара, на рисунке 56 для спроса, на рисунке 57 для величины поставок, на рисунке 58 для макропеременной.

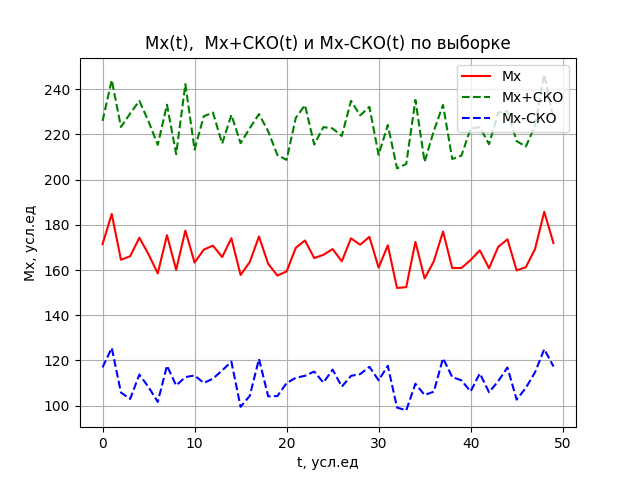


Рисунок 55 – График с учётом СКО по выборке

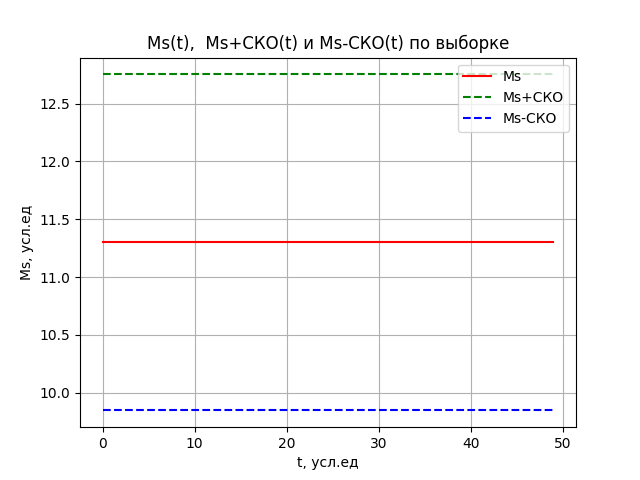


Рисунок 56 – График с учётом СКО по выборке

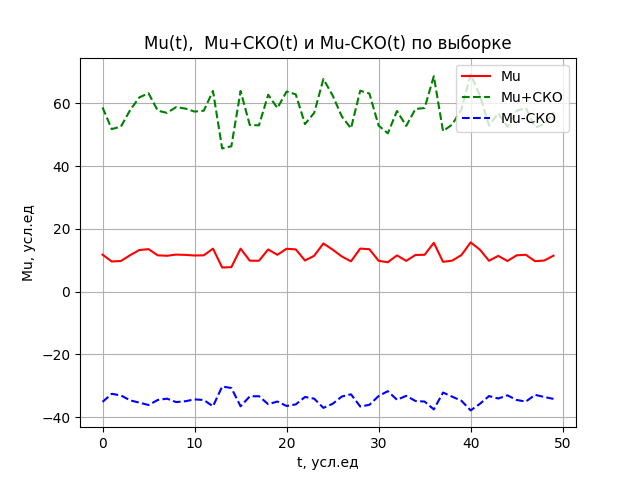
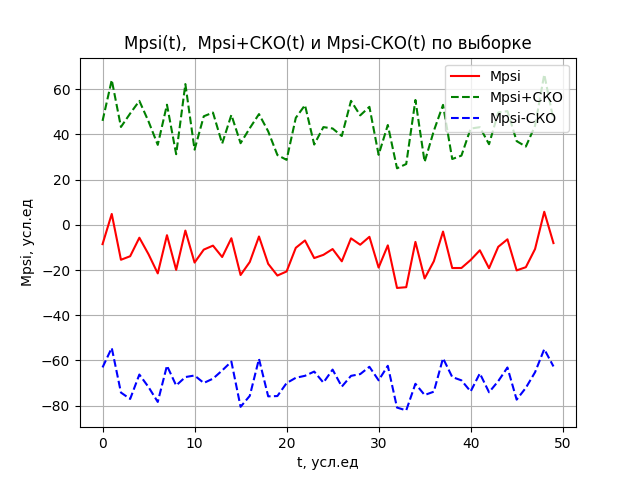


Рисунок 57 – График с учётом СКО по выборке

 Рисунок 58 – График с учётом СКО по выборке

На рисунке 59 представлен график зависимости количества затраченной энергии на обеспечение управления от времени для математической модели, описанной выше.

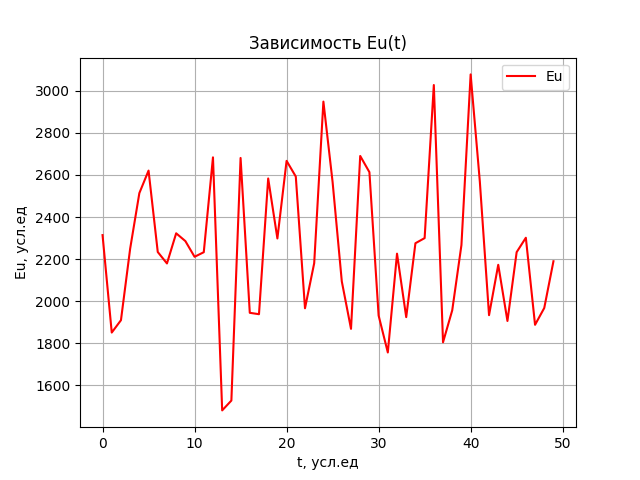


Рисунок 59 – График зависимости (t)

Графики оценок математического ожидания с учётом СКО по 50 ансамблям для каждой из исследуемых переменных модели, описанной выше, представлены на рисунке 60 для количества товара, на рисунке 61 для спроса, на рисунке 62 для величины поставок, на рисунке 63 для макропеременной.

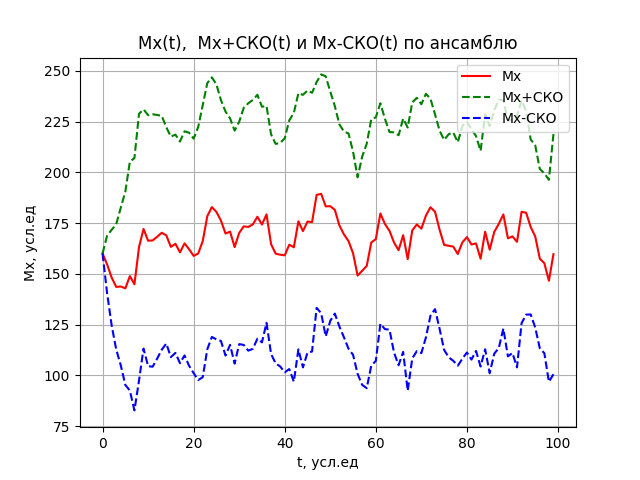


Рисунок 60 – График з с учётом СКО по ансамблю

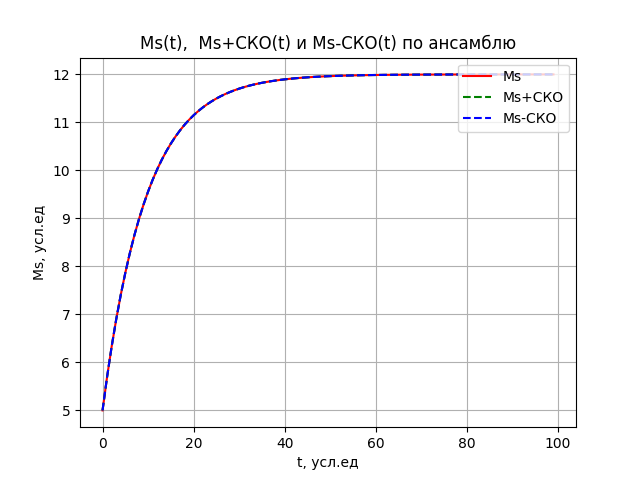


Рисунок 61 – График с учётом СКО по ансамблю

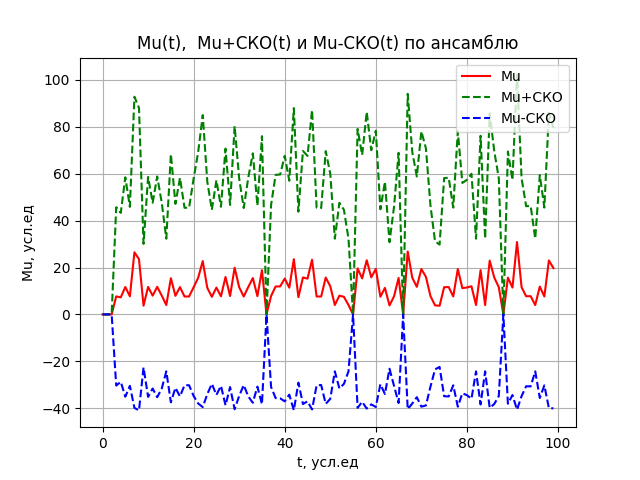


Рисунок 62 – График с учётом СКО по ансамблю

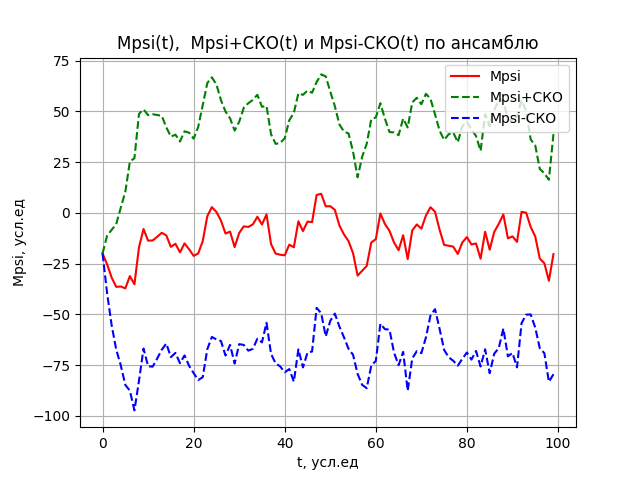


Рисунок 63 – График с учётом СКО по ансамблю

## 5.3 Анализ влияния параметра регулятора на модель

В блоке 4 программы исследовалось влияние параметра регулятора на результаты моделирования. Исследование проводилось для модели с параметрами: и значениями параметра регулятора . Результаты моделирования представлены на графике зависимости количества товара от времени на рисунке 64, на графике зависимости спроса от времени на рисунке 65, на графике зависимости величины поставок от времени на рисунке 66, на графике зависимости макропеременной от времени на рисунке 67.

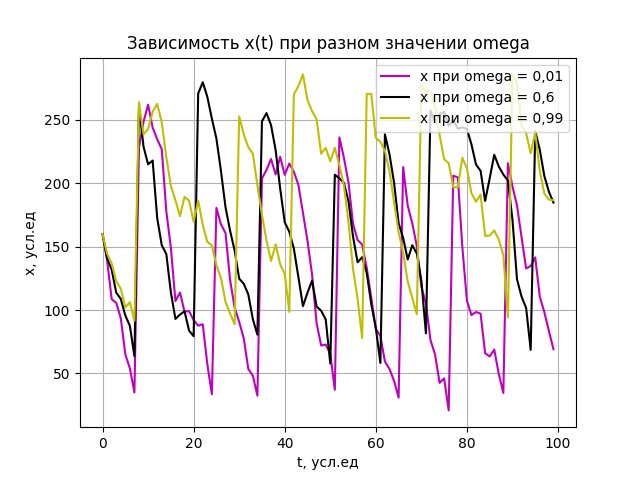


Рисунок 64 – График зависимости x(t)

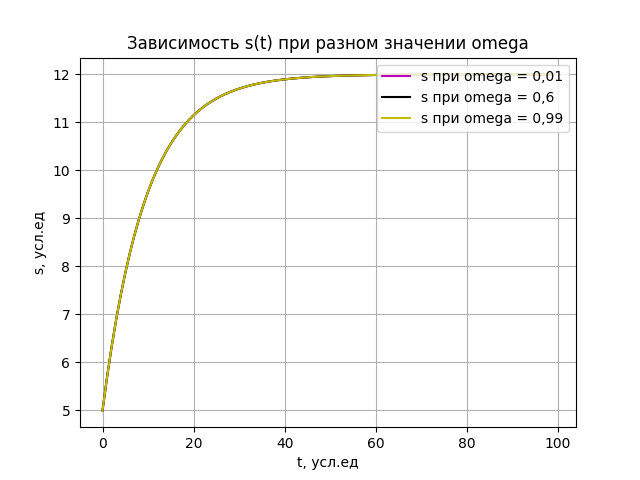


Рисунок 65 – График зависимости s(t)

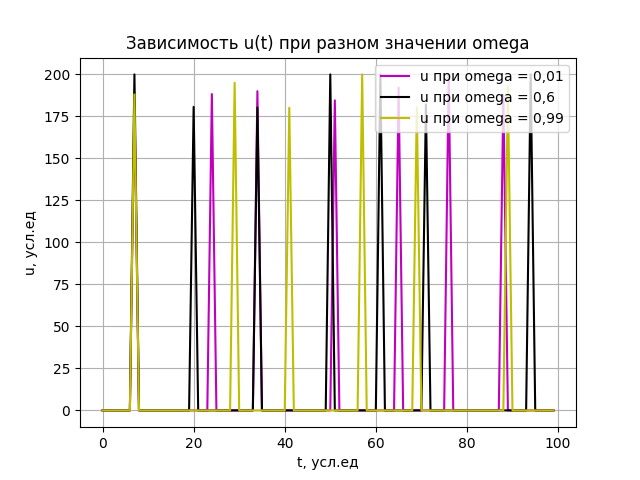


Рисунок 66 – График зависимости u(t)

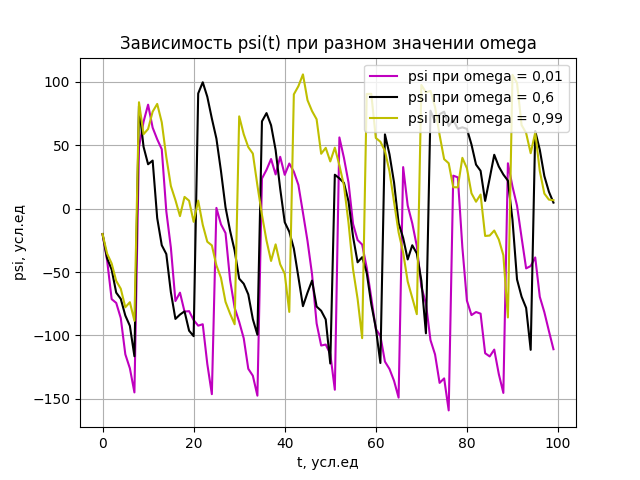


Рисунок 67 – График зависимости (t)

По полученным графика можно сделать следующий вывод: при любом значении параметра регулятора однопродуктовый склад работает, что говорит об устойчивости модели, разница между результатами заключается лишь в величине поставок.

## 5.4 Анализ влияния начальных условий на модель

В блоке 5 программы исследовалось влияние начальных условий на результаты моделирования. Исследование проводилось для модели с параметрами: , и начальными условиями . Результаты моделирования представлены на графике зависимости количества товара от времени на рисунке 68, на графике зависимости спроса от времени на рисунке 69, на графике зависимости величины поставок от времени на рисунке 70, на графике зависимости макропеременной от времени на рисунке 71.

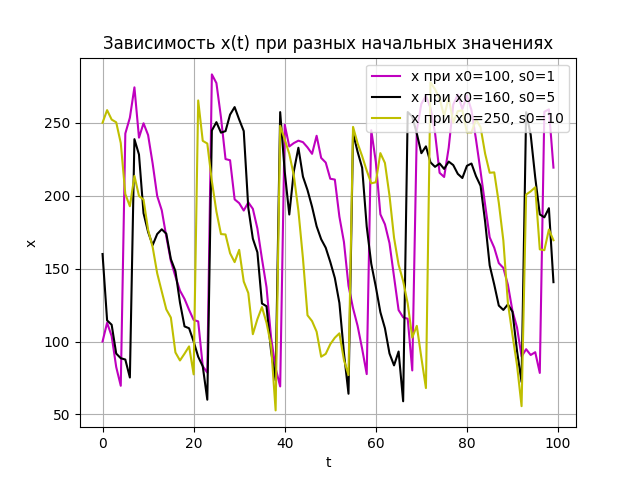


Рисунок 68 – График зависимости x(t)

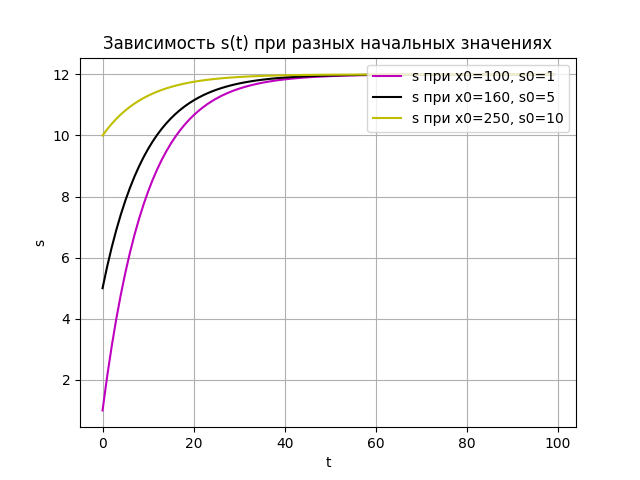


Рисунок 69 – График зависимости s(t)

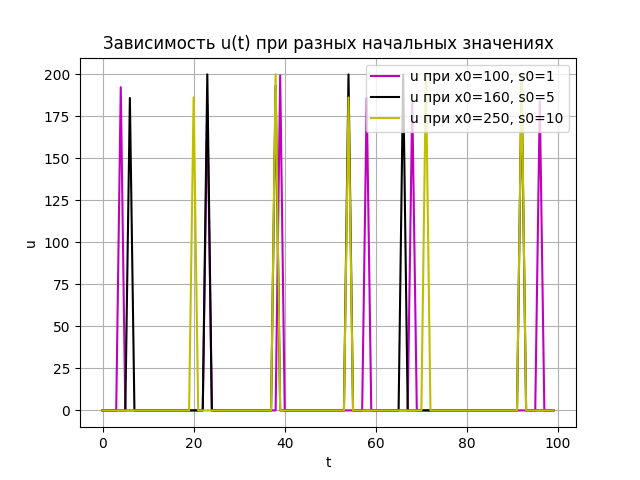


Рисунок 70 – График зависимости u(t)



Рисунок 71 – График зависимости (t)

По полученным графика можно сделать следующий вывод: при любом значении начальных параметров склад работает, что говорит об устойчивости модели, разница между результатами заключается лишь в величине поставок.

## 5.5 Итоговая таблица

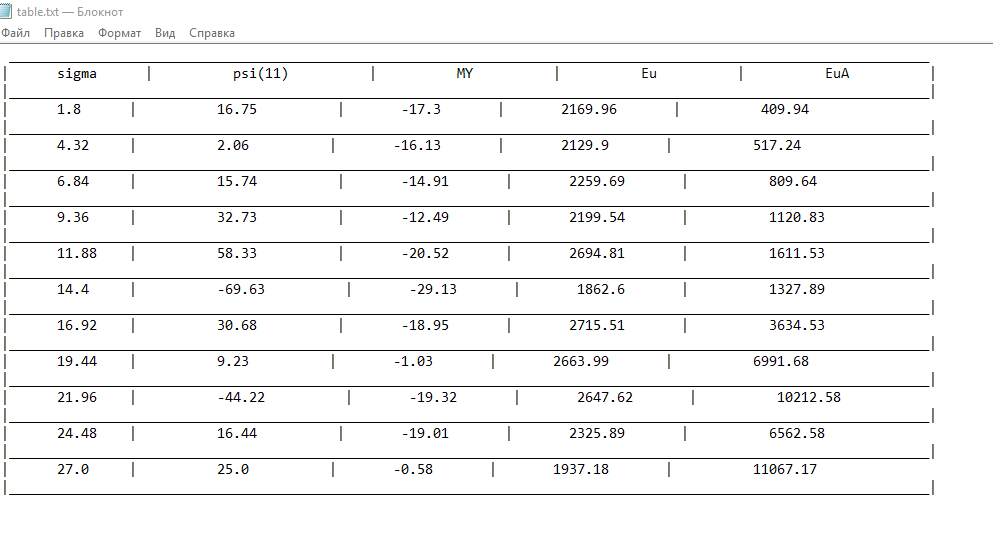


Рисунок 72 – Таблица, предоставляемая пользователю, после окончания расчётов модели

# **ЗАКЛЮЧЕНИЕ**

Цель исследования – применение метода аналитического конструирования агрегированных регуляторов (АКАР) для оптимизации однопродуктового склада – успешно достигнута.

В ходе выполнения работы были решены следующие задачи.

1. Получены формулы для управления на основе метода аналитического конструирования агрегированных регуляторов (АКАР) для оптимизации работы однопродуктового склада в условиях стохастической неопределённости;
2. Разработано программное обеспечение для тестирования полученной модели и её вариаций;
3. Проведено исследование зависимости модели от параметра регулятора и исходных данных.

Теоретическая значимость: новая система управления для модели управления однопродуктового склада в условиях стохастической неопределённости.

Практическая значимость: приложение для проведения численных экспериментов с моделью, позволяющее пользователю для исходных данных спрогнозировать частотность и объём поставок.

# **СПИСОК ЛИТЕРАТУРЫ**

* 1. Хэнссмен Ф. Применение математических методов в управлении производством и запасами /Ф. Хэнссмен. — Москва: Прогресс, 1966.
  2. Прабху Н. Стохастические процессы теории запасов / Н. Прахбу — Москва: Мир, 1984.
  3. Колесникова C.И. Синтез стохастического регулятора на многообразиях для нелинейного дискретного объекта // Прикладная информатика. 2018. № 6. С. 18–30.
  4. Кolesnikova S.I. Multiple-control system for nonlinear stochastic discrete object / Optimization Methods and Software. Vol.34, Issue 3. 2018. с. 578-585
  5. Колесников А.А. Синергетика и проблемы теории управления: сборник научных трудов/ А.А. Колесников — Москва: Физматлит, 2004.
  6. Смагин В.И. Локально-оптимальное управление запасами. Учебно-методическое пособие. / В.И. Смагин — Томск: ТГУ, 2001.
  7. Колесников А.А., Колесников А. А., Кузьменко А.А. Методы АКАР и АКОР в задачах синтеза нелинейных систем управления / Мехатроника, автоматизация, управление — Москва: Новые технологии, 2016.
  8. Лутц М. Изучаем Python, 4-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2011. – 1280 с.
  9. Прохоренок Н.А. Python 3 и PyQt. Разработка приложений. – СПб.: БХВ-Петербург, 2012. – 704 с.
  10. Доусон М. Программируем на Python. – СПб.: Питер, 2014. – 416 с.
  11. Информационные технологии и вычислительные системы: Высокопроизводительные вычислительные системы. Математическое моделирование. Методы обработки информации / Под ред. С.В. Емельянова. - М.: Ленанд, 2012. - 100 c.
  12. Труды ИСА РАН: Алгоритмы. Решения. Математическое моделирование. Управление рисками и безопасностью / Под ред. С.В. Емельянова. - М.: Ленанд, 2014. - 102 c.
  13. Пестунов, М. А., Маркова, Н. В. Экономическая роль товарных запасов в логистической системе российских предприятий / М.А. Пестунов, Н.В. Маркова — Молодой ученый. — 2018. — №16. — С. 189-191.
  14. Шрайбфедер Дж. Эффективное управление запасами/Дж. Шрайбфедер; пер. с англ. Ю. Орлова. Изд-во Альпина Паблишер, 2019. 304 с.
  15. Федоров, Д. Ю. Программирование на языке высокого уровня Python : учебное пособие для прикладного бакалавриата / Д. Ю. Федоров. – 2-е изд., перераб. и доп. – Москва : Издательство Юрайт, 2019. – 161 с. – (Бакалавр. Прикладной курс). – ISBN 978-5-534-10971-9. – Текст: электронный // ЭБС Юрайт [сайт]. – URL: https://urait.ru/bcode/437489 (дата обращения: 13.02.2020).
  16. Рейтц К., Шлюссер Т. Автостопом по Python. – СПб.: Питер, 2017. – 336 с.: ил. – (Серия «Бестселлеры O’Reilly»).
  17. Шелудько, В. М. Язык программирования высокого уровня Python. Функции, структуры данных, дополнительные модули: учебное пособие / В. М. Шелудько. – Ростов-на-Дону, Таганрог: Издательство Южного федерального университета, 2017. – 107 c.
  18. Python documentation // 2023. URL: <https://www.python.org/doc/> (дата обращения: 13.05.2023
  19. Алексеева, Н.И., Алёхина, Ю.В. Управление товарными запасами предприятия на основе анализа системы показателей / Н.И. Алексеева, Ю.В. Алёхина // Стратегия предприятия в контексте повышения его конкурентоспособности. 2019. Т. 1. № 5 (5). С. 56-59.
  20. Грызанов, Ю.П., Файницкий, А.И. Управление товарными запасами в торговле./ Ю.П. Грызанов, А.И. Файницкий — М.: Экономика, 2019. -215 с.
  21. Сазонова, А.К., Матвийчук, Л.Н. Управление запасами предприятия и их оптимизация / А.К. Сазонова, Л.Н. Матвийчук // Современные тенденции развития науки и технологий. 2019. Т. 8. № 5. С. 124-127.

# **ПРИЛОЖЕНИЕ**

|  |
| --- |
| import sys  import matplotlib.pyplot as plt import numpy as np import math  import os from PyQt5 import uic from PyQt5.QtWidgets import QApplication, QMessageBox  *# !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! # Расчетные функции # Поиск математического ожидания* def find\_m\_t(list\_p):  list\_res = []  for num in range(len(list\_p)):  summa\_mas = sum(list\_p[num])  *# Находим мат ожидание* m = summa\_mas / len(list\_p[num])  list\_res.append(m)  return list\_res   *# Поиск дисперсии* def find\_d\_t(list\_p, list\_m):  list\_res = []  for num in range(len(list\_p)):  summa\_mas = 0  for ch in range(len(list\_p[num])):  summa\_mas += (list\_p[num][ch] - list\_m[num]) \*\* 2  *# Находим дисперсию* D = summa\_mas / (len(list\_p[num]) - 1)  list\_res.append(D)  return list\_res   *# Поиск СКО* def find\_sigma\_t(list\_d):  list\_sigma = []  for num in range(len(list\_d)):  sig = math.sqrt(list\_d[num])  list\_sigma.append(sig)  return list\_sigma   *# Поиск ОШС* def find\_SNR(const, list\_for\_sig, list\_for\_all\_p, list\_m):  list\_SNR = []  for num in range(len(list\_m)):  sum\_mas = 0  for ch in range(len(list\_for\_all\_p[num])):  sum\_mas += list\_for\_all\_p[num][ch] \*\* 2  number1 = (1 + const) \*\* 2  number2 = list\_for\_sig[num] \*\* 2  number3 = (sum\_mas / len(list\_for\_all\_p[num])) - list\_m[num] \*\* 2  list\_SNR.append(number1 \* number2 / number3)  return list\_SNR   *# Найдем мат ожидание минус СКО* def find\_M\_minus\_sigma(list\_M, list\_sigma):  list\_M\_min\_sig = []  for num in range(len(list\_M)):  list\_M\_min\_sig.append(list\_M[num] - list\_sigma[num])  return list\_M\_min\_sig   *# Найдем мат ожидание плюс СКО* def find\_M\_plus\_sigma(list\_M, list\_sigma):  list\_M\_pl\_sig = []  for num in range(len(list\_M)):  list\_M\_pl\_sig.append(list\_M[num] + list\_sigma[num])  return list\_M\_pl\_sig   *# Найдем мат ожидание по ансамблю* def find\_M\_ans(list\_p):  list\_M\_ans = []  for num in range(len(list\_p[0])):  summa\_ans = 0  for ch in range(len(list\_p)):  summa\_ans += list\_p[ch][num]  M\_ans = summa\_ans / len(list\_p)  list\_M\_ans.append(M\_ans)  return list\_M\_ans   *# Найдем дисперсию по ансамблю* def find\_D\_ans(list\_p, list\_M):  list\_D\_ans = []  for num in range(len(list\_p[0])):  summa\_ans = 0  for ch in range(len(list\_p)):  summa\_ans += (list\_p[ch][num] - list\_M[num]) \*\* 2  D\_ans = summa\_ans / (len(list\_p) - 1)  list\_D\_ans.append(D\_ans)   return list\_D\_ans   *# Функция для поиска энергии управления* def find\_energy\_u(list\_per):  list\_e = []  for num in range(len(list\_per)):  sum\_u = 0  for ch in range(len(list\_per[num])):  sum\_u += list\_per[num][ch] \*\* 2  list\_e.append(sum\_u / len(list\_per[num]))  return list\_e  def for\_rounnd\_zn(list\_zn):  list\_r = []  for i in range(len(list\_zn)):  list\_r.append(round((list\_zn[i]), 3))  return list\_r  *# !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! # Функции для отрисовки # Строим графики зависимости x, s, u от Т* def draw\_graph\_perem\_t(list\_x, list\_s, list\_u, list\_Y):  *# #Рисуем графики* list\_for\_t = []  for num in range(len(list\_x[0])):  list\_for\_t.append(num)   for ch in range(11):  fig = plt.figure()  plt.plot(list\_for\_t, list\_x[ch], 'ro')  plt.title('Зависимость x(t)')  plt.ylabel('x, усл.ед')  plt.xlabel('t, усл.ед')  plt.grid(True)  plt.legend(['x'], loc=1)  fig.savefig('Pic\_block1\_' + str(ch + 1) + '\_for\_x\_t.png')  plt.close()  fig\_1 = plt.figure()  plt.plot(list\_for\_t, list\_s[ch], 'ro')  plt.title('Зависимость s(t)')  plt.ylabel('s, усл.ед')  plt.xlabel('t, усл.ед')  plt.grid(True)  plt.legend(['s'], loc=1)  *# plt.show()* fig\_1.savefig('Pic\_block1\_' + str(ch + 1) + '\_for\_s\_t.png')  plt.close()  fig\_2 = plt.figure()  plt.plot(list\_for\_t, list\_u[ch], 'ro')  plt.title('Зависимость u(t)')  plt.ylabel('u, усл.ед')  plt.xlabel('t, усл.ед')  plt.grid(True)  plt.legend(['u'], loc=1)  *# plt.show()* fig\_2.savefig('Pic\_block1\_' + str(ch + 1) + '\_for\_Y\_t.png')  plt.close()  fig\_2 = plt.figure()  plt.plot(list\_for\_t, list\_Y[ch], 'ro')  plt.title('Зависимость psi(t)')  plt.ylabel('psi, усл.ед')  plt.xlabel('t, усл.ед')  plt.grid(True)  plt.legend(['psi'], loc=1)  *# plt.show()* fig\_2.savefig('Pic\_block1\_' + str(ch + 1) + '\_for\_psi\_t.png')  plt.close()   *# Построим графики зависимости оценок мат ожидания от ОШС* def draw\_graph\_mark\_SNR(list\_snr\_x, list\_m\_x, list\_snr\_s, list\_m\_s, list\_snr\_u, list\_m\_u, list\_snr\_Y, list\_m\_Y, list\_r\_x\_snr, list\_r\_s\_snr, list\_r\_u\_snr, list\_r\_Y\_snr ):  fig\_snr\_1 = plt.figure()  plt.plot(list\_snr\_x, list\_m\_x, 'r')  plt.title('Зависимость Mx(ОШС))')  plt.ylabel('Mx, усл.ед')  plt.xlabel('ОШС, усл.ед')  plt.grid(True)  plt.legend(['Mx'], loc=1)  fig\_snr\_1.savefig('Pic\_block1\_Mx\_snr.png')  plt.close()   fig\_snr\_1 = plt.figure()  plt.plot(list\_snr\_x, list\_m\_x, 'r.')  plt.title('Зависимость Mx(ОШС)')  plt.ylabel('Mx, усл.ед')  plt.xlabel('ОШС, усл.ед')  plt.grid(True)  plt.legend(['Mx'], loc=1)  fig\_snr\_1.savefig('Pic\_block1\_Mx\_snr\_2.png')  plt.close()   fig\_snr\_2 = plt.figure()  plt.plot(list\_snr\_s, list\_m\_s, 'r')  plt.title('Зависимость Ms(ОШС)')  plt.ylabel('Ms, усл.ед')  plt.xlabel('ОШС, усл.ед')  plt.grid(True)  plt.legend(['Ms'], loc=1)  fig\_snr\_2.savefig('Pic\_block1\_Ms\_snr.png')  plt.close()   fig\_snr\_2 = plt.figure()  plt.plot(list\_snr\_s, list\_m\_s, 'r.')  plt.title('Зависимость Ms(ОШС)')  plt.ylabel('Ms, усл.ед')  plt.xlabel('ОШС, усл.ед')  plt.grid(True)  plt.legend(['Ms'], loc=1)  fig\_snr\_2.savefig('Pic\_block1\_Ms\_snr\_2.png')  plt.close()   fig\_snr\_3 = plt.figure()  plt.plot(list\_snr\_u, list\_m\_u, 'r')  plt.title('Зависимость Mu(ОШС)')  plt.ylabel('Mu, усл.ед')  plt.xlabel('ОШС, усл.ед')  plt.grid(True)  plt.legend(['Mu'], loc=1)  fig\_snr\_3.savefig('Pic\_block1\_Mu\_snr.png')  plt.close()   fig\_snr\_3 = plt.figure()  plt.plot(list\_snr\_u, list\_m\_u, 'r.')  plt.title('Зависимость Mu(ОШС))')  plt.ylabel('Mu, усл.ед')  plt.xlabel('ОШС, усл.ед')  plt.grid(True)  plt.legend(['Mu'], loc=1)  fig\_snr\_3.savefig('Pic\_block1\_Mu\_snr\_2.png')  plt.close()   fig\_snr\_4 = plt.figure()  plt.plot(list\_snr\_Y, list\_m\_Y, 'r')  plt.title('Зависимость Mpsi(ОШС))')  plt.ylabel('Mpsi, усл.ед')  plt.xlabel('ОШС, усл.ед')  plt.grid(True)  plt.legend(['Mpsi'], loc=1)  fig\_snr\_4.savefig('Pic\_block1\_MY\_snr.png')  plt.close()   fig\_snr\_4 = plt.figure()  plt.plot(list\_snr\_Y, list\_m\_Y, 'r.')  plt.title('Зависимость Mpsi(ОШС))')  plt.ylabel('Mpsi, усл.ед')  plt.xlabel('ОШС, усл.ед')  plt.grid(True)  plt.legend(['Mpsi'], loc=1)  fig\_snr\_4.savefig('Pic\_block1\_MY\_snr\_2.png')  plt.close()  *# Построим график зависимости энергии управления от ОШС* def draw\_graph\_Eu(list\_snr, list\_e):  *# Построим график зависимости Энергии управления от ОШС* fig\_snr = plt.figure()  plt.plot(list\_snr,list\_e, color="red")  plt.title('Зависимость Eu(ОШС)')  plt.ylabel('Eu, усл.ед')  plt.xlabel('ОШС, усл.ед')  plt.grid(True)  plt.legend(['Eu'], loc=1)  fig\_snr.savefig('Pic\_block1\_E\_u\_snr.png')  plt.close()   def print\_perem (a, b, c, w, x\_, Umin, Umax, s, x, u, sigma\_const ):  *# \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ БЛОК 1 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  # Целевая макропеременная в 0 момент времени* Y = x - x\_   *# Создадим списк для списков данных x, s, u, Y, sigma* list\_for\_all\_x = []  list\_for\_all\_s = []  list\_for\_all\_u = []  list\_for\_all\_Y = []  list\_for\_sigma = [x\_ \* 0.01, x\_ \* 0.024, x\_ \* 0.038, x\_ \* 0.052, x\_ \* 0.066, x\_ \* 0.08, x\_ \* 0.094, x\_ \* 0.108, x\_ \* 0.122, x\_ \* 0.136, x\_ \* 0.150]   *# list\_for\_sigma = [1.6, 1.6, 1.6, 1.6, 1.6, 1.6, 1.6, 1.6, 1.6, 1.6, 1.6]* for n in range(11):  sigma = list\_for\_sigma[n]  *# значение шума в 0 момент времени* e = np.random.normal(0, float(sigma), 1)  *# формируем список значений шума* list\_for\_e = [float(e)]  *# формируем список значений кол-ва товаров на складе* list\_for\_x = [float(x)]  *# формируем список значений величины спроса* list\_for\_s = [float(s)]  *# формируем список значений объема поставок* list\_for\_u = [float(u)]  *# формируем список значений макропеременной* list\_for\_Y = [float(Y)]  for t in range(1, 100):  *# получаем значение спроса в момент времени t и добавляем его в список* s\_t = a \* list\_for\_s[t - 1] + b  list\_for\_s.append(float(s\_t))  *# получаем новое значение шума и добавляем его в список* e\_t = np.random.normal(0, float(sigma), 1)  list\_for\_e.append(float(e\_t))  *# получаем новое значение кол-ва товаров на складе в момент времени t и добавляем его в список* x\_t = list\_for\_x[t - 1] + list\_for\_u[t - 1] + c \* list\_for\_e[t - 1] + e\_t - s\_t  list\_for\_x.append(float(x\_t))  *# получем значение макропеременной и добавляем его в список* Y\_t = x\_t - x\_  list\_for\_Y.append(float(Y\_t))  *# получаем момент спроса в t+1* s\_t\_plus\_1 = a \* s\_t + b  *# получим значение управления по методу акар* u\_a = (w + c) \* (-Y\_t) + x\_ + c \* w \* list\_for\_Y[t - 1] + s\_t\_plus\_1 - x\_t   if u\_a < Umin:  *# получаем объём поставок в момент времени t* u\_t = 0  list\_for\_u.append(float(u\_t))  elif u\_a < Umax:  u\_t = u\_a  list\_for\_u.append(float(u\_t))  else:  u\_t = Umax  list\_for\_u.append(float(u\_t))  *# добавляем полученные данные в массив с массивами* list\_for\_all\_x.append(list\_for\_x)  list\_for\_all\_s.append(list\_for\_s)  list\_for\_all\_u.append(list\_for\_u)  list\_for\_all\_Y.append(list\_for\_Y)  *# Выводим зачения на экран* print('x: ', list\_for\_x)  print('s: ', list\_for\_s)  print('u: ', list\_for\_u)  print('Y: ', list\_for\_Y)   *# Поиск оценок на временном промежутке  # формируем список для найденных оценок мат ожиданий* list\_for\_M\_x = find\_m\_t(list\_for\_all\_x)  list\_for\_M\_s = find\_m\_t(list\_for\_all\_s)  list\_for\_M\_u = find\_m\_t(list\_for\_all\_u)  list\_for\_M\_Y = find\_m\_t(list\_for\_all\_Y)  print('M\_x', list\_for\_M\_x)  print('M\_s', list\_for\_M\_s)  print('M\_u', list\_for\_M\_u)  print('M\_Y', list\_for\_M\_Y)   *# формируем список для найденных дисперсий* list\_for\_D\_x = find\_d\_t(list\_for\_all\_x, list\_for\_M\_x)  list\_for\_D\_s = find\_d\_t(list\_for\_all\_s, list\_for\_M\_s)  list\_for\_D\_u = find\_d\_t(list\_for\_all\_u, list\_for\_M\_u)  list\_for\_D\_Y = find\_d\_t(list\_for\_all\_Y, list\_for\_M\_Y)  print('D\_x', list\_for\_D\_x)  print('D\_s', list\_for\_D\_s)  print('D\_u', list\_for\_D\_u)  print('D\_Y', list\_for\_D\_Y)   *# формируем список для найденных СКО* list\_for\_sigma\_x = find\_sigma\_t(list\_for\_D\_x)  list\_for\_sigma\_s = find\_sigma\_t(list\_for\_D\_s)  list\_for\_sigma\_u = find\_sigma\_t(list\_for\_D\_u)  list\_for\_sigma\_Y = find\_sigma\_t(list\_for\_D\_Y)  print('sigma\_x', list\_for\_sigma\_x)  print('sigma\_s', list\_for\_sigma\_s)  print('sigma\_u', list\_for\_sigma\_u)  print('sigma\_Y', list\_for\_sigma\_Y)   *# формируем список для ОШС x* list\_for\_SNR\_x = find\_SNR(c, list\_for\_sigma, list\_for\_all\_x, list\_for\_M\_x)  list\_for\_SNR\_s = find\_SNR(c, list\_for\_sigma, list\_for\_all\_s, list\_for\_M\_s)  list\_for\_SNR\_u = find\_SNR(c, list\_for\_sigma, list\_for\_all\_u, list\_for\_M\_u)  list\_for\_SNR\_Y = find\_SNR(c, list\_for\_sigma, list\_for\_all\_Y, list\_for\_M\_Y)  print('SNR\_x', list\_for\_SNR\_x)  print('SNR\_s', list\_for\_SNR\_s)  print('SNR\_u', list\_for\_SNR\_u)  print('SNR\_Y', list\_for\_SNR\_Y)   *# округлим значения для графиков* list\_for\_SNR\_x\_r = for\_rounnd\_zn(list\_for\_SNR\_x)  list\_for\_SNR\_s\_r = for\_rounnd\_zn(list\_for\_SNR\_s)  list\_for\_SNR\_u\_r = for\_rounnd\_zn(list\_for\_SNR\_u)  list\_for\_SNR\_Y\_r = for\_rounnd\_zn(list\_for\_SNR\_Y)   *# формируем список для энергии* list\_for\_energy\_u = find\_energy\_u(list\_for\_all\_u)  print('energy\_u', list\_for\_energy\_u)   *# Посчитиаем значения мат ожиданий с учетом СКО для графиков* list\_for\_M\_x\_minus\_sigma = find\_M\_minus\_sigma(list\_for\_M\_x, list\_for\_sigma\_x)  list\_for\_M\_x\_plus\_sigma = find\_M\_plus\_sigma(list\_for\_M\_x, list\_for\_sigma\_x)  list\_for\_M\_s\_minus\_sigma = find\_M\_minus\_sigma(list\_for\_M\_s, list\_for\_sigma\_s)  list\_for\_M\_s\_plus\_sigma = find\_M\_plus\_sigma(list\_for\_M\_s, list\_for\_sigma\_s)  list\_for\_M\_u\_minus\_sigma = find\_M\_minus\_sigma(list\_for\_M\_u, list\_for\_sigma\_u)  list\_for\_M\_u\_plus\_sigma = find\_M\_plus\_sigma(list\_for\_M\_u, list\_for\_sigma\_u)  list\_for\_M\_Y\_minus\_sigma = find\_M\_minus\_sigma(list\_for\_M\_Y, list\_for\_sigma\_Y)  list\_for\_M\_Y\_plus\_sigma = find\_M\_plus\_sigma(list\_for\_M\_Y, list\_for\_sigma\_Y)   *# Выведем значения на экран* print('list\_for\_M\_x\_plus\_sigma', list\_for\_M\_x\_plus\_sigma)  print('list\_for\_M\_x\_minus\_sigma', list\_for\_M\_x\_minus\_sigma)  print('list\_for\_M\_s\_plus\_sigma', list\_for\_M\_s\_plus\_sigma)  print('list\_for\_M\_s\_minus\_sigma', list\_for\_M\_s\_minus\_sigma)  print('list\_for\_M\_u\_plus\_sigma', list\_for\_M\_u\_plus\_sigma)  print('list\_for\_M\_u\_minus\_sigma', list\_for\_M\_u\_minus\_sigma)  print('list\_for\_M\_Y\_plus\_sigma', list\_for\_M\_Y\_plus\_sigma)  print('list\_for\_M\_Y\_minus\_sigma', list\_for\_M\_Y\_minus\_sigma)   *# Построим графики x, s, u от t* draw\_graph\_perem\_t(list\_for\_all\_x, list\_for\_all\_s, list\_for\_all\_u, list\_for\_all\_Y)  *# Построим графики зависимости оценок мат ожидания от ОШС* draw\_graph\_mark\_SNR(list\_for\_SNR\_x, list\_for\_M\_x, list\_for\_SNR\_s, list\_for\_M\_s, list\_for\_SNR\_u, list\_for\_M\_u,  list\_for\_SNR\_Y, list\_for\_M\_Y, list\_for\_SNR\_x\_r, list\_for\_SNR\_s\_r, list\_for\_SNR\_u\_r,  list\_for\_SNR\_Y\_r)  *# Построим график зависимости Энергии управления от ОШС* draw\_graph\_Eu(list\_for\_SNR\_u, list\_for\_energy\_u)   *# Построим графики* fig\_snr = plt.figure()  plt.plot(list\_for\_SNR\_x, list\_for\_M\_x, 'r', list\_for\_SNR\_x, list\_for\_M\_x\_plus\_sigma, 'g--', list\_for\_SNR\_x,  list\_for\_M\_x\_minus\_sigma, 'b--')  plt.title('Зависимость Mx(ОШС), Mx+СКО(ОШС) и Mx-СКО(ОШС)')  plt.ylabel('Mx, усл.ед')  plt.xlabel('ОШС, усл.ед')  plt.grid(True)  plt.legend(['Mx', 'Mx+СКО', 'Mx-СКО'], loc=1)  fig\_snr.savefig('Pic\_block1\_Mx\_snr\_sigma.png')  plt.close()   fig\_snr = plt.figure()  plt.plot(list\_for\_SNR\_s, list\_for\_M\_s, 'r', list\_for\_SNR\_s, list\_for\_M\_s\_plus\_sigma, 'g--', list\_for\_SNR\_s,  list\_for\_M\_s\_minus\_sigma, 'b--')  plt.title('Зависимость Ms(ОШС), Ms+СКО(ОШС) и Ms-СКО(ОШС)')  plt.ylabel('Ms, усл.ед')  plt.xlabel('ОШС, усл.ед')  plt.grid(True)  plt.legend(['Ms', 'Ms+СКО', 'Ms-СКО'], loc=1)  fig\_snr.savefig('Pic\_block1\_Ms\_snr\_sigma.png')  plt.close()   fig\_snr = plt.figure()  plt.plot(list\_for\_SNR\_u, list\_for\_M\_u, 'r', list\_for\_SNR\_u, list\_for\_M\_u\_plus\_sigma, 'g--', list\_for\_SNR\_u,  list\_for\_M\_u\_minus\_sigma, 'b--')  plt.title('Зависимость Mu(ОШС), Mu+СКО(ОШС) и Mu-СКО(ОШС)')  plt.ylabel('Mu, усл.ед')  plt.xlabel('ОШС, усл.ед')  plt.grid(True)  plt.legend(['Mu', 'Mu+СКО', 'Mu-СКО'], loc=1)  fig\_snr.savefig('Pic\_block1\_Mu\_snr\_sigma.png')  plt.close()   fig\_snr = plt.figure()  plt.plot(list\_for\_SNR\_Y, list\_for\_M\_Y, 'r', list\_for\_SNR\_Y, list\_for\_M\_Y\_plus\_sigma, 'g--', list\_for\_SNR\_Y,  list\_for\_M\_Y\_minus\_sigma, 'b--')  plt.title('Зависимость Mpsi(ОШС), Mpsi+СКО(ОШС) и Mpsi-СКО(ОШС)')  plt.ylabel('Mpsi, усл.ед')  plt.xlabel('ОШС, усл.ед')  plt.grid(True)  plt.legend(['Mpsi', 'Mpsi+СКО', 'Mpsi-СКО'], loc=1)  fig\_snr.savefig('Pic\_block1\_MY\_snr\_sigma.png')  plt.close()   *# \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ БЛОК 2 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  # ЧАСТЬ БЕЗ ОГРАНИЧЕНИЙ ДЛЯ УПРАВЛЕНИЯ  # Создадим списк для списков данных x, s, u, Y, sigma* list\_for\_all\_x\_2 = []  list\_for\_all\_s\_2 = []  list\_for\_all\_u\_2 = []  list\_for\_all\_Y\_2 = []  for n in range(11):  sigma = list\_for\_sigma[n]  *# значение шума в 0 момент времени* e = np.random.normal(0, float(sigma), 1)  *# формируем список значений шума* list\_for\_e = [float(e)]  *# формируем список значений кол-ва товаров на складе* list\_for\_x = [float(x)]  *# формируем список значений величины спроса* list\_for\_s = [float(s)]  *# формируем список значений объема поставок* list\_for\_u = [float(u)]  *# формируем список значений макропеременной* list\_for\_Y = [float(Y)]  for t in range(1, 100):  *# получаем значение спроса в момент времени t и добавляем его в список* s\_t = a \* list\_for\_s[t - 1] + b  list\_for\_s.append(float(s\_t))  *# получаем новое значение шума и добавляем его в список* e\_t = np.random.normal(0, float(sigma), 1)  list\_for\_e.append(float(e\_t))  *# получаем новое значение кол-ва товаров на складе в момент времени t и добавляем его в список* x\_t = list\_for\_x[t - 1] + list\_for\_u[t - 1] + c \* list\_for\_e[t - 1] + e\_t - s\_t  list\_for\_x.append(float(x\_t))  *# получем значение макропеременной и добавляем его в список* Y\_t = x\_t - x\_  list\_for\_Y.append(float(Y\_t))  *# получаем момент спроса в t+1* s\_t\_plus\_1 = a \* s\_t + b  *# получим значение управления по методу акар* u\_a = (w + c) \* (-Y\_t) + x\_ + c \* w \* list\_for\_Y[t - 1] + s\_t\_plus\_1 - x\_t  u\_t = u\_a  list\_for\_u.append(float(u\_t))   *# добавляем полученные данные в массив с массивами* list\_for\_all\_x\_2.append(list\_for\_x)  list\_for\_all\_s\_2.append(list\_for\_s)  list\_for\_all\_u\_2.append(list\_for\_u)  list\_for\_all\_Y\_2.append(list\_for\_Y)  *# print('x\_2: ', list\_for\_x)  # print('s\_2: ', list\_for\_s)  # print('u\_2: ', list\_for\_u)  # print('Y\_2: ', list\_for\_Y)* print('list\_for\_all\_x\_2', list\_for\_all\_x\_2)  print('list\_for\_all\_s\_2', list\_for\_all\_s\_2)  print('list\_for\_all\_u\_2', list\_for\_all\_u\_2)  print('list\_for\_all\_Y\_2', list\_for\_all\_Y\_2)   *# #Рисуем графики* list\_for\_t = []  for num in range(len(list\_for\_all\_x\_2[0])):  list\_for\_t.append(num)   for t in range(11):  fig = plt.figure()  plt.plot(list\_for\_t, list\_for\_all\_x[t], 'r.', list\_for\_t, list\_for\_all\_x\_2[t], 'b.')  plt.title('Зависимость x(t) с ограничением и x2(t) без ограничения')  plt.ylabel('x, усл.ед')  plt.xlabel('t, усл.ед')  plt.grid(True)  plt.legend(['x', 'x2'], loc=1)  fig.savefig('Pic\_block2\_' + str(t + 1) + '\_for\_x\_t\_and\_x\_2\_t.png')  plt.close()  fig\_1 = plt.figure()  plt.plot(list\_for\_t, list\_for\_all\_s[t], 'r.', list\_for\_t, list\_for\_all\_s\_2[t], 'b.')  plt.title('Зависимость s(t) с ограничением и s2(t) без ограничения')  plt.ylabel('s, усл.ед')  plt.xlabel('t, усл.ед')  plt.grid(True)  plt.legend(['s', 's2'], loc=1)  *# plt.show()* fig\_1.savefig('Pic\_block2\_' + str(t + 1) + '\_for\_s\_t\_and\_s\_2\_t.png')  plt.close()  fig\_2 = plt.figure()  plt.plot(list\_for\_t, list\_for\_all\_u[t], 'r.', list\_for\_t, list\_for\_all\_u\_2[t], 'b.')  plt.title('Зависимость u(t) с ограничением и u2(t) без ограничения')  plt.ylabel('u, усл.ед')  plt.xlabel('t, усл.ед')  plt.grid(True)  plt.legend(['u', 'u2'], loc=1)  *# plt.show()* fig\_2.savefig('Pic\_block2\_' + str(t + 1) + '\_for\_u\_t\_and\_u\_2\_t.png')  plt.close()  fig\_2 = plt.figure()  plt.plot(list\_for\_t, list\_for\_all\_Y[t], 'r.', list\_for\_t, list\_for\_all\_Y\_2[t], 'b.')  plt.title('Зависимость psi(t) с ограничением и psi2(t) без ограничения')  plt.ylabel('psi, усл.ед')  plt.xlabel('t, усл.ед')  plt.grid(True)  plt.legend(['psi', 'psi2'], loc=1)  *# plt.show()* fig\_2.savefig('Pic\_block2\_' + str(t + 1) + '\_for\_Y\_t\_and\_Y\_2\_t.png')  plt.close()   *# Посчитаем оценки для модели без ограничений на управление  # формируем спсисок для найденных оценок мат ожиданий* list\_for\_M\_x\_2 = find\_m\_t(list\_for\_all\_x\_2)  list\_for\_M\_s\_2 = find\_m\_t(list\_for\_all\_s\_2)  list\_for\_M\_u\_2 = find\_m\_t(list\_for\_all\_u\_2)  list\_for\_M\_Y\_2 = find\_m\_t(list\_for\_all\_Y\_2)  print('M\_x\_2', list\_for\_M\_x\_2)  print('M\_s\_2', list\_for\_M\_s\_2)  print('M\_u\_2', list\_for\_M\_u\_2)  print('M\_Y\_2', list\_for\_M\_Y\_2)   *# формируем список для найденных дисперсий* list\_for\_D\_x\_2 = find\_d\_t(list\_for\_all\_x\_2, list\_for\_M\_x\_2)  list\_for\_D\_s\_2 = find\_d\_t(list\_for\_all\_s\_2, list\_for\_M\_s\_2)  list\_for\_D\_u\_2 = find\_d\_t(list\_for\_all\_u\_2, list\_for\_M\_u\_2)  list\_for\_D\_Y\_2 = find\_d\_t(list\_for\_all\_Y\_2, list\_for\_M\_Y\_2)  print('D\_x\_2', list\_for\_D\_x\_2)  print('D\_s\_2', list\_for\_D\_s\_2)  print('D\_u\_2', list\_for\_D\_u\_2)  print('D\_Y\_2', list\_for\_D\_Y\_2)   *# формируем список для найденных СКО* list\_for\_sigma\_x\_2 = find\_sigma\_t(list\_for\_D\_x\_2)  list\_for\_sigma\_s\_2 = find\_sigma\_t(list\_for\_D\_s\_2)  list\_for\_sigma\_u\_2 = find\_sigma\_t(list\_for\_D\_u\_2)  list\_for\_sigma\_Y\_2 = find\_sigma\_t(list\_for\_D\_Y\_2)  print('sigma\_x\_2', list\_for\_sigma\_x\_2)  print('sigma\_s\_2', list\_for\_sigma\_s\_2)  print('sigma\_u\_2', list\_for\_sigma\_u\_2)  print('sigma\_Y\_2', list\_for\_sigma\_Y\_2)   *# формируем список для ОШС x* list\_for\_SNR\_x\_2 = find\_SNR(c, list\_for\_sigma, list\_for\_all\_x\_2, list\_for\_M\_x\_2)  list\_for\_SNR\_s\_2 = find\_SNR(c, list\_for\_sigma, list\_for\_all\_s\_2, list\_for\_M\_s\_2)  list\_for\_SNR\_u\_2 = find\_SNR(c, list\_for\_sigma, list\_for\_all\_u\_2, list\_for\_M\_u\_2)  list\_for\_SNR\_Y\_2 = find\_SNR(c, list\_for\_sigma, list\_for\_all\_Y\_2, list\_for\_M\_Y\_2)  print('SNR\_x\_2', list\_for\_SNR\_x\_2)  print('SNR\_s\_2', list\_for\_SNR\_s\_2)  print('SNR\_u\_2', list\_for\_SNR\_u\_2)  print('SNR\_Y\_2', list\_for\_SNR\_Y\_2)   fig\_snr = plt.figure()  plt.plot(list\_for\_SNR\_x\_2, list\_for\_M\_x\_2, color="red")  plt.title('Зависимость Mx(ОШС))')  plt.ylabel('Mx, усл.ед')  plt.xlabel('ОШС, усл.ед')  plt.grid(True)  plt.legend(['Mx'], loc=1)  fig\_snr.savefig('Pic\_block2\_Mx\_snr\_2.png')  plt.close()   fig\_snr = plt.figure()  plt.plot(list\_for\_SNR\_s\_2, list\_for\_M\_s\_2, color="red")  plt.title('Зависимость Ms(ОШС)')  plt.ylabel('Ms, усл.ед')  plt.xlabel('ОШС, усл.ед')  plt.grid(True)  plt.legend(['Ms'], loc=1)  fig\_snr.savefig('Pic\_block2\_Ms\_snr\_2.png')  plt.close()   fig\_snr = plt.figure()  plt.plot(list\_for\_SNR\_u\_2, list\_for\_M\_u\_2, color="red")  plt.title('Зависимость Mu(ОШС)')  plt.ylabel('Mu, усл.ед')  plt.xlabel('ОШС, усл.ед')  plt.grid(True)  plt.legend(['Mu'], loc=1)  fig\_snr.savefig('Pic\_block2\_Mu\_snr\_2.png')  plt.close()   fig\_snr = plt.figure()  plt.plot(list\_for\_SNR\_Y\_2, list\_for\_M\_Y\_2, color="red")  plt.title('Зависимость Mpsi(ОШС))')  plt.ylabel('Mpsi, усл.ед')  plt.xlabel('ОШС, усл.ед')  plt.grid(True)  plt.legend(['Mpsi'], loc=1)  fig\_snr.savefig('Pic\_block2\_MY\_snr\_2.png')  plt.close()   *# формируем список для энергии* list\_for\_energy\_u\_2 = find\_energy\_u(list\_for\_all\_u\_2)  print('energy\_u\_2', list\_for\_energy\_u\_2)   *# Рисуем график энергии* fig\_snr = plt.figure()  plt.plot(list\_for\_SNR\_u\_2, list\_for\_energy\_u\_2, color="red")  plt.title('Зависимость Eu(ОШС))')  plt.ylabel('Eu, усл.ед')  plt.xlabel('ОШС, усл.ед')  plt.grid(True)  plt.legend(['Eu(ОШС)'], loc=1)  fig\_snr.savefig('Pic\_block2\_E\_u\_snr\_2.png')  plt.close()   list\_for\_M\_x\_minus\_sigma\_2 = find\_M\_minus\_sigma(list\_for\_M\_x\_2, list\_for\_sigma\_x\_2)  list\_for\_M\_x\_plus\_sigma\_2 = find\_M\_plus\_sigma(list\_for\_M\_x\_2, list\_for\_sigma\_x\_2)  list\_for\_M\_s\_minus\_sigma\_2 = find\_M\_minus\_sigma(list\_for\_M\_s\_2, list\_for\_sigma\_s\_2)  list\_for\_M\_s\_plus\_sigma\_2 = find\_M\_plus\_sigma(list\_for\_M\_s\_2, list\_for\_sigma\_s\_2)  list\_for\_M\_u\_minus\_sigma\_2 = find\_M\_minus\_sigma(list\_for\_M\_u\_2, list\_for\_sigma\_u\_2)  list\_for\_M\_u\_plus\_sigma\_2 = find\_M\_plus\_sigma(list\_for\_M\_u\_2, list\_for\_sigma\_u\_2)  list\_for\_M\_Y\_minus\_sigma\_2 = find\_M\_minus\_sigma(list\_for\_M\_Y\_2, list\_for\_sigma\_Y\_2)  list\_for\_M\_Y\_plus\_sigma\_2 = find\_M\_plus\_sigma(list\_for\_M\_Y\_2, list\_for\_sigma\_Y\_2)   print('list\_for\_M\_x\_plus\_sigma\_2', list\_for\_M\_x\_plus\_sigma\_2)  print('list\_for\_M\_x\_minus\_sigma\_2', list\_for\_M\_x\_minus\_sigma\_2)  print('list\_for\_M\_s\_plus\_sigma\_2', list\_for\_M\_s\_plus\_sigma\_2)  print('list\_for\_M\_s\_minus\_sigma\_2', list\_for\_M\_s\_minus\_sigma\_2)  print('list\_for\_M\_u\_plus\_sigma\_2', list\_for\_M\_u\_plus\_sigma\_2)  print('list\_for\_M\_u\_minus\_sigma\_2', list\_for\_M\_u\_minus\_sigma\_2)  print('list\_for\_M\_Y\_plus\_sigma\_2', list\_for\_M\_Y\_plus\_sigma\_2)  print('list\_for\_M\_Y\_minus\_sigma\_2', list\_for\_M\_Y\_minus\_sigma\_2)   *# Построим графики* fig\_snr = plt.figure()  plt.plot(list\_for\_SNR\_x\_2, list\_for\_M\_x\_2, 'r', list\_for\_SNR\_x\_2, list\_for\_M\_x\_plus\_sigma\_2, 'g--',  list\_for\_SNR\_x\_2, list\_for\_M\_x\_minus\_sigma\_2, 'b--')  plt.title('Зависимость Mx(ОШС), Mx+СКО(ОШС) и Mx-СКО(ОШС)')  plt.ylabel('Mx, усл.ед')  plt.xlabel('ОШС, усл.ед')  plt.grid(True)  plt.legend(['Mx', 'Mx+СКО', 'Mx-СКО'], loc=1)  fig\_snr.savefig('Pic\_block2\_Mx\_snr\_sigma\_2.png')  plt.close()   fig\_snr = plt.figure()  plt.plot(list\_for\_SNR\_s\_2, list\_for\_M\_s\_2, 'r', list\_for\_SNR\_s\_2, list\_for\_M\_s\_plus\_sigma\_2, 'g--',  list\_for\_SNR\_s\_2, list\_for\_M\_s\_minus\_sigma\_2, 'b--')  plt.title('Зависимость Ms(ОШС), Ms+СКО(ОШС) и Ms-СКО(ОШС)')  plt.ylabel('Ms, усл.ед')  plt.xlabel('ОШС, усл.ед')  plt.grid(True)  plt.legend(['Ms', 'Ms+СКО', 'Ms-СКО'], loc=1)  fig\_snr.savefig('Pic\_block2\_Ms\_snr\_sigma\_2.png')  plt.close()   fig\_snr = plt.figure()  plt.plot(list\_for\_SNR\_u\_2, list\_for\_M\_u\_2, 'r', list\_for\_SNR\_u\_2, list\_for\_M\_u\_plus\_sigma\_2, 'g--',  list\_for\_SNR\_u\_2, list\_for\_M\_u\_minus\_sigma\_2, 'b--')  plt.title('Зависимость Mu(ОШС), Mu+СКО(ОШС) и Mu-СКО(ОШС)')  plt.ylabel('Mu, усл.ед')  plt.xlabel('ОШС, усл.ед')  plt.grid(True)  plt.legend(['Mu', 'Mu+СКО', 'Mu-СКО'], loc=1)  fig\_snr.savefig('Pic\_block2\_Mu\_snr\_sigma\_2.png')  plt.close()   fig\_snr = plt.figure()  plt.plot(list\_for\_SNR\_Y\_2, list\_for\_M\_Y\_2, 'r', list\_for\_SNR\_Y\_2, list\_for\_M\_Y\_plus\_sigma\_2, 'g--',  list\_for\_SNR\_Y\_2, list\_for\_M\_Y\_minus\_sigma\_2, 'b--')  plt.title('Зависимость Mpsi(ОШС), Mpsi+СКО(ОШС) и Mpsi-СКО(ОШС)')  plt.ylabel('Mpsi, усл.ед')  plt.xlabel('ОШС, усл.ед')  plt.grid(True)  plt.legend(['Mpsi', 'Mpsi+СКО', 'Mpsi-СКО'], loc=1)  fig\_snr.savefig('Pic\_block2\_MY\_snr\_sigma\_2.png')  plt.close()   *# \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ БЛОК 3 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  # С одинаковой сигмой  # Создадим списк для списков данных x, s, u, Y, sigma* list\_for\_all\_x\_sigma\_const = []  list\_for\_all\_s\_sigma\_const = []  list\_for\_all\_u\_sigma\_const = []  list\_for\_all\_Y\_sigma\_const = []  for n in range(50):  e = np.random.normal(0, float(list\_for\_sigma[5]), 1)  *# формируем список значений шума* list\_for\_e = [float(e)]  *# формируем список значений кол-ва товаров на складе* list\_for\_x = [float(x)]  *# формируем список значений величины спроса* list\_for\_s = [float(s)]  *# формируем список значений объема поставок* list\_for\_u = [float(u)]  *# формируем список значений макропеременной* list\_for\_Y = [float(Y)]  for t in range(1, 100):  *# получаем значение спроса в момент времени t и добавляем его в список* s\_t = a \* list\_for\_s[t - 1] + b  list\_for\_s.append(float(s\_t))  *# получаем новое значение шума и добавляем его в список* e\_t = np.random.normal(0, float(list\_for\_sigma[5]), 1)  list\_for\_e.append(float(e\_t))  *# получаем новое значение кол-ва товаров на складе в момент времени t и добавляем его в список* x\_t = list\_for\_x[t - 1] + list\_for\_u[t - 1] + c \* list\_for\_e[t - 1] + e\_t - s\_t  list\_for\_x.append(float(x\_t))  *# получем значение макропеременной и добавляем его в список* Y\_t = x\_t - x\_  list\_for\_Y.append(float(Y\_t))  *# получаем момент спроса в t+1* s\_t\_plus\_1 = a \* s\_t + b  *# получим значение управления по методу акар* u\_a = (w + c) \* (-Y\_t) + x\_ + c \* w \* list\_for\_Y[t - 1] + s\_t\_plus\_1 - x\_t   if u\_a < Umin:  *# получаем объём поставок в момент времени t* u\_t = 0  list\_for\_u.append(float(u\_t))  elif u\_a < Umax:  u\_t = u\_a  list\_for\_u.append(float(u\_t))  else:  u\_t = Umax  list\_for\_u.append(float(u\_t))  *# добавляем полученные данные в массив с массивами* list\_for\_all\_x\_sigma\_const.append(list\_for\_x)  list\_for\_all\_s\_sigma\_const.append(list\_for\_s)  list\_for\_all\_u\_sigma\_const.append(list\_for\_u)  list\_for\_all\_Y\_sigma\_const.append(list\_for\_Y)  print('list\_for\_x\_sigma\_const', list\_for\_x)  print('list\_for\_s\_sigma\_const', list\_for\_s)  print('list\_for\_u\_sigma\_const', list\_for\_u)  print('list\_for\_Y\_sigma\_const', list\_for\_Y)   *# #Рисуем графики* list\_for\_t\_const = []  for t in range(100):  list\_for\_t\_const.append(t)   for t in range(11):  fig = plt.figure()  plt.plot(list\_for\_t\_const, list\_for\_all\_x\_sigma\_const[t], 'ro')  plt.title('Зависимость x(t)')  plt.ylabel('x, усл.ед')  plt.xlabel('t, усл.ед')  plt.grid(True)  plt.legend(['x'], loc=1)  fig.savefig('Pic\_block3\_' + str(t + 1) + '\_for\_x\_t\_sigma\_const.png')  plt.close()  fig\_1 = plt.figure()  plt.plot(list\_for\_t\_const, list\_for\_all\_s\_sigma\_const[t], 'ro')  plt.title('Зависимость s(t)')  plt.ylabel('s, усл.ед')  plt.xlabel('t, усл.ед')  plt.grid(True)  plt.legend(['s'], loc=1)  *# plt.show()* fig\_1.savefig('Pic\_block3\_' + str(t + 1) + '\_for\_s\_t\_sigma\_const.png')  plt.close()  fig\_2 = plt.figure()  plt.plot(list\_for\_t\_const, list\_for\_all\_u\_sigma\_const[t], 'ro')  plt.title('Зависимость u(t)')  plt.ylabel('u, усл.ед')  plt.xlabel('t, усл.ед')  plt.grid(True)  plt.legend(['u'], loc=1)  *# plt.show()* fig\_2.savefig('Pic\_block3\_' + str(t + 1) + '\_for\_u\_t\_sigma\_const.png')  plt.close()  fig\_2 = plt.figure()  plt.plot(list\_for\_t\_const, list\_for\_all\_Y\_sigma\_const[t], 'ro')  plt.title('Зависимость psi(t)')  plt.ylabel('psi, усл.ед')  plt.xlabel('t, усл.ед')  plt.grid(True)  plt.legend(['psi'], loc=1)  *# plt.show()* fig\_2.savefig('Pic\_block3\_' + str(t + 1) + '\_for\_Y\_t\_sigma\_const.png')  plt.close()   *# Поиск оценок на временном промежутке  # формируем спсисок для найденных оценок мат ожиданий* list\_for\_M\_x\_sigma\_const = find\_m\_t(list\_for\_all\_x\_sigma\_const)  list\_for\_M\_s\_sigma\_const = find\_m\_t(list\_for\_all\_s\_sigma\_const)  list\_for\_M\_u\_sigma\_const = find\_m\_t(list\_for\_all\_u\_sigma\_const)  list\_for\_M\_Y\_sigma\_const = find\_m\_t(list\_for\_all\_Y\_sigma\_const)  print('M\_x\_sigma\_const', list\_for\_M\_x\_sigma\_const)  print('M\_s\_sigma\_const', list\_for\_M\_s\_sigma\_const)  print('M\_u\_sigma\_const', list\_for\_M\_u\_sigma\_const)  print('M\_Y\_sigma\_const', list\_for\_M\_Y\_sigma\_const)   *# формируем список для найденных дисперсий* list\_for\_D\_x\_sigma\_const = find\_d\_t(list\_for\_all\_x\_sigma\_const, list\_for\_M\_x\_sigma\_const)  list\_for\_D\_s\_sigma\_const = find\_d\_t(list\_for\_all\_s\_sigma\_const, list\_for\_M\_s\_sigma\_const)  list\_for\_D\_u\_sigma\_const = find\_d\_t(list\_for\_all\_u\_sigma\_const, list\_for\_M\_u\_sigma\_const)  list\_for\_D\_Y\_sigma\_const = find\_d\_t(list\_for\_all\_Y\_sigma\_const, list\_for\_M\_Y\_sigma\_const)  print('D\_x\_sigma\_const', list\_for\_D\_x\_sigma\_const)  print('D\_s\_sigma\_const', list\_for\_D\_s\_sigma\_const)  print('D\_u\_sigma\_const', list\_for\_D\_u\_sigma\_const)  print('D\_Y\_sigma\_const', list\_for\_D\_Y\_sigma\_const)   *# формируем список для найденных СКО* list\_for\_sigma\_x\_sigma\_const = find\_sigma\_t(list\_for\_D\_x\_sigma\_const)  list\_for\_sigma\_s\_sigma\_const = find\_sigma\_t(list\_for\_D\_s\_sigma\_const)  list\_for\_sigma\_u\_sigma\_const = find\_sigma\_t(list\_for\_D\_u\_sigma\_const)  list\_for\_sigma\_Y\_sigma\_const = find\_sigma\_t(list\_for\_D\_Y\_sigma\_const)  print('sigma\_x\_sigma\_const', list\_for\_sigma\_x\_sigma\_const)  print('sigma\_s\_sigma\_const', list\_for\_sigma\_s\_sigma\_const)  print('sigma\_u\_sigma\_const', list\_for\_sigma\_u\_sigma\_const)  print('sigma\_Y\_sigma\_const', list\_for\_sigma\_Y\_sigma\_const)   list\_for\_M\_x\_minus\_sigma\_const = find\_M\_minus\_sigma(list\_for\_M\_x\_sigma\_const, list\_for\_sigma\_x\_sigma\_const)  list\_for\_M\_x\_plus\_sigma\_const = find\_M\_plus\_sigma(list\_for\_M\_x\_sigma\_const, list\_for\_sigma\_x\_sigma\_const)  list\_for\_M\_s\_minus\_sigma\_const = find\_M\_minus\_sigma(list\_for\_M\_s\_sigma\_const, list\_for\_sigma\_s\_sigma\_const)  list\_for\_M\_s\_plus\_sigma\_const = find\_M\_plus\_sigma(list\_for\_M\_s\_sigma\_const, list\_for\_sigma\_s\_sigma\_const)  list\_for\_M\_u\_minus\_sigma\_const = find\_M\_minus\_sigma(list\_for\_M\_u\_sigma\_const, list\_for\_sigma\_u\_sigma\_const)  list\_for\_M\_u\_plus\_sigma\_const = find\_M\_plus\_sigma(list\_for\_M\_u\_sigma\_const, list\_for\_sigma\_u\_sigma\_const)  list\_for\_M\_Y\_minus\_sigma\_const = find\_M\_minus\_sigma(list\_for\_M\_Y\_sigma\_const, list\_for\_sigma\_Y\_sigma\_const)  list\_for\_M\_Y\_plus\_sigma\_const = find\_M\_plus\_sigma(list\_for\_M\_Y\_sigma\_const, list\_for\_sigma\_Y\_sigma\_const)   print('list\_for\_M\_x\_plus\_sigma\_const', list\_for\_M\_x\_plus\_sigma\_const)  print('list\_for\_M\_x\_minus\_sigma\_const', list\_for\_M\_x\_minus\_sigma\_const)  print('list\_for\_M\_s\_plus\_sigma\_const', list\_for\_M\_s\_plus\_sigma\_const)  print('list\_for\_M\_s\_minus\_sigma\_const', list\_for\_M\_s\_minus\_sigma\_const)  print('list\_for\_M\_u\_plus\_sigma\_const', list\_for\_M\_u\_plus\_sigma\_const)  print('list\_for\_M\_u\_minus\_sigma\_const', list\_for\_M\_u\_minus\_sigma\_const)  print('list\_for\_M\_Y\_plus\_sigma\_const', list\_for\_M\_Y\_plus\_sigma\_const)  print('list\_for\_M\_Y\_minus\_sigma\_const', list\_for\_M\_Y\_minus\_sigma\_const)   *# Рисуем графики* list\_for\_T = []  for T in range(50):  list\_for\_T.append(T)  fig\_snr = plt.figure()   plt.plot(list\_for\_T, list\_for\_M\_x\_sigma\_const, 'r', list\_for\_T, list\_for\_M\_x\_plus\_sigma\_const, 'g--', list\_for\_T,  list\_for\_M\_x\_minus\_sigma\_const, 'b--')  plt.title('Mx(t), Mx+СКО(t) и Mx-СКО(t) по выборке')  plt.ylabel('Mx, усл.ед')  plt.xlabel('t, усл.ед')  plt.grid(True)  plt.legend(['Mx', 'Mx+СКО', 'Mx-СКО'], loc=1)  fig\_snr.savefig('Pic\_block3\_Mx\_t\_sigma\_sigma\_const.png')   fig\_snr = plt.figure()  plt.plot(list\_for\_T, list\_for\_M\_s\_sigma\_const, 'r', list\_for\_T, list\_for\_M\_s\_plus\_sigma\_const, 'g--', list\_for\_T,  list\_for\_M\_s\_minus\_sigma\_const, 'b--')  plt.title('Ms(t), Ms+СКО(t) и Ms-СКО(t) по выборке')  plt.ylabel('Ms, усл.ед')  plt.xlabel('t, усл.ед')  plt.grid(True)  plt.legend(['Ms', 'Ms+СКО', 'Ms-СКО'], loc=1)  fig\_snr.savefig('Pic\_block3\_Ms\_t\_sigma\_sigma\_const.png')   fig\_snr = plt.figure()  plt.plot(list\_for\_T, list\_for\_M\_u\_sigma\_const, 'r', list\_for\_T, list\_for\_M\_u\_plus\_sigma\_const, 'g--', list\_for\_T,  list\_for\_M\_u\_minus\_sigma\_const, 'b--')  plt.title('Mu(t), Mu+СКО(t) и Mu-СКО(t) по выборке')  plt.ylabel('Mu, усл.ед')  plt.xlabel('t, усл.ед')  plt.grid(True)  plt.legend(['Mu', 'Mu+СКО', 'Mu-СКО'], loc=1)  fig\_snr.savefig('Pic\_block3\_Mu\_t\_sigma\_sigma\_const.png')   fig\_snr = plt.figure()  plt.plot(list\_for\_T, list\_for\_M\_Y\_sigma\_const, 'r', list\_for\_T, list\_for\_M\_Y\_plus\_sigma\_const, 'g--', list\_for\_T,  list\_for\_M\_Y\_minus\_sigma\_const, 'b--')  plt.title('Mpsi(t), Mpsi+СКО(t) и Mpsi-СКО(t) по выборке')  plt.ylabel('Mpsi, усл.ед')  plt.xlabel('t, усл.ед')  plt.grid(True)  plt.legend(['Mpsi', 'Mpsi+СКО', 'Mpsi-СКО'], loc=1)  fig\_snr.savefig('Pic\_block3\_MY\_t\_sigma\_sigma\_const.png')   *# Поиск оценок по ансамблю  # формируем спсисок для найденных оценок мат ожиданий* list\_for\_M\_x\_ans = find\_M\_ans(list\_for\_all\_x\_sigma\_const)  list\_for\_M\_s\_ans = find\_M\_ans(list\_for\_all\_s\_sigma\_const)  list\_for\_M\_u\_ans = find\_M\_ans(list\_for\_all\_u\_sigma\_const)  list\_for\_M\_Y\_ans = find\_M\_ans(list\_for\_all\_Y\_sigma\_const)   print('M\_x\_ans', list\_for\_M\_x\_ans)  print('M\_s\_ans', list\_for\_M\_s\_ans)  print('M\_u\_ans', list\_for\_M\_u\_ans)  print('M\_Y\_ans', list\_for\_M\_Y\_ans)   *# формируем список для найденных дисперсий по ансамблю* list\_for\_D\_x\_ans = find\_D\_ans(list\_for\_all\_x\_sigma\_const, list\_for\_M\_x\_ans)  list\_for\_D\_s\_ans = find\_D\_ans(list\_for\_all\_s\_sigma\_const, list\_for\_M\_s\_ans)  list\_for\_D\_u\_ans = find\_D\_ans(list\_for\_all\_u\_sigma\_const, list\_for\_M\_u\_ans)  list\_for\_D\_Y\_ans = find\_D\_ans(list\_for\_all\_Y\_sigma\_const, list\_for\_M\_Y\_ans)  print('D\_x\_ans', list\_for\_D\_x\_ans)  print('D\_s\_ans', list\_for\_D\_s\_ans)  print('D\_u\_ans', list\_for\_D\_u\_ans)  print('D\_Y\_ans', list\_for\_D\_Y\_ans)   *# формируем список для найденных СКО по ансамблю* list\_for\_sigma\_x\_ans = find\_sigma\_t(list\_for\_D\_x\_ans)  list\_for\_sigma\_s\_ans = find\_sigma\_t(list\_for\_D\_s\_ans)  list\_for\_sigma\_u\_ans = find\_sigma\_t(list\_for\_D\_u\_ans)  list\_for\_sigma\_Y\_ans = find\_sigma\_t(list\_for\_D\_Y\_ans)  print('sigma\_x\_ans', list\_for\_sigma\_x\_ans)  print('sigma\_s\_ans', list\_for\_sigma\_s\_ans)  print('sigma\_u\_ans', list\_for\_sigma\_u\_ans)  print('sigma\_Y\_ans', list\_for\_sigma\_Y\_ans)   list\_for\_M\_x\_minus\_sigma\_ans = find\_M\_minus\_sigma(list\_for\_M\_x\_ans, list\_for\_sigma\_x\_ans)  list\_for\_M\_x\_plus\_sigma\_ans = find\_M\_plus\_sigma(list\_for\_M\_x\_ans, list\_for\_sigma\_x\_ans)  list\_for\_M\_s\_minus\_sigma\_ans = find\_M\_minus\_sigma(list\_for\_M\_s\_ans, list\_for\_sigma\_s\_ans)  list\_for\_M\_s\_plus\_sigma\_ans = find\_M\_plus\_sigma(list\_for\_M\_s\_ans, list\_for\_sigma\_s\_ans)  list\_for\_M\_u\_minus\_sigma\_ans = find\_M\_minus\_sigma(list\_for\_M\_u\_ans, list\_for\_sigma\_u\_ans)  list\_for\_M\_u\_plus\_sigma\_ans = find\_M\_plus\_sigma(list\_for\_M\_u\_ans, list\_for\_sigma\_u\_ans)  list\_for\_M\_Y\_minus\_sigma\_ans = find\_M\_minus\_sigma(list\_for\_M\_Y\_ans, list\_for\_sigma\_Y\_ans)  list\_for\_M\_Y\_plus\_sigma\_ans = find\_M\_plus\_sigma(list\_for\_M\_Y\_ans, list\_for\_sigma\_Y\_ans)   print('list\_for\_M\_x\_plus\_sigma\_ans', list\_for\_M\_x\_plus\_sigma\_ans)  print('list\_for\_M\_x\_minus\_sigma\_ans', list\_for\_M\_x\_minus\_sigma\_ans)  print('list\_for\_M\_s\_plus\_sigma\_ans', list\_for\_M\_s\_plus\_sigma\_ans)  print('list\_for\_M\_s\_minus\_sigma\_ans', list\_for\_M\_s\_minus\_sigma\_ans)  print('list\_for\_M\_u\_plus\_sigma\_ans', list\_for\_M\_u\_plus\_sigma\_ans)  print('list\_for\_M\_u\_minus\_sigma\_ans', list\_for\_M\_u\_minus\_sigma\_ans)  print('list\_for\_M\_Y\_plus\_sigma\_ans', list\_for\_M\_Y\_plus\_sigma\_ans)  print('list\_for\_M\_Y\_minus\_sigma\_ans', list\_for\_M\_Y\_minus\_sigma\_ans)   *# Рисуем графики* list\_for\_T\_ans = []  for T in range(100):  list\_for\_T\_ans.append(T)  fig\_snr = plt.figure()   plt.plot(list\_for\_T\_ans, list\_for\_M\_x\_ans, 'r', list\_for\_T\_ans, list\_for\_M\_x\_plus\_sigma\_ans, 'g--', list\_for\_T\_ans,  list\_for\_M\_x\_minus\_sigma\_ans, 'b--')  plt.title('Mx(t), Mx+СКО(t) и Mx-СКО(t) по ансамблю')  plt.ylabel('Mx, усл.ед')  plt.xlabel('t, усл.ед')  plt.grid(True)  plt.legend(['Mx', 'Mx+СКО', 'Mx-СКО'], loc=1)  fig\_snr.savefig('Pic\_block3\_Mx\_ans\_sigma\_sigma\_const.png')   fig\_snr = plt.figure()  plt.plot(list\_for\_T\_ans, list\_for\_M\_s\_ans, 'r', list\_for\_T\_ans, list\_for\_M\_s\_plus\_sigma\_ans, 'g--', list\_for\_T\_ans,  list\_for\_M\_s\_minus\_sigma\_ans, 'b--')  plt.title('Ms(t), Ms+СКО(t) и Ms-СКО(t) по ансамблю')  plt.ylabel('Ms, усл.ед')  plt.xlabel('t, усл.ед')  plt.grid(True)  plt.legend(['Ms', 'Ms+СКО', 'Ms-СКО'], loc=1)  fig\_snr.savefig('Pic\_block3\_Ms\_ans\_sigma\_sigma\_const.png')   fig\_snr = plt.figure()  plt.plot(list\_for\_T\_ans, list\_for\_M\_u\_ans, 'r', list\_for\_T\_ans, list\_for\_M\_u\_plus\_sigma\_ans, 'g--', list\_for\_T\_ans,  list\_for\_M\_u\_minus\_sigma\_ans, 'b--')  plt.title('Mu(t), Mu+СКО(t) и Mu-СКО(t) по ансамблю')  plt.ylabel('Mu, усл.ед')  plt.xlabel('t, усл.ед')  plt.grid(True)  plt.legend(['Mu', 'Mu+СКО', 'Mu-СКО'], loc=1)  fig\_snr.savefig('Pic\_block3\_Mu\_ans\_sigma\_sigma\_const.png')   fig\_snr = plt.figure()  plt.plot(list\_for\_T\_ans, list\_for\_M\_Y\_ans, 'r', list\_for\_T\_ans, list\_for\_M\_Y\_plus\_sigma\_ans, 'g--', list\_for\_T\_ans,  list\_for\_M\_Y\_minus\_sigma\_ans, 'b--')  plt.title('Mpsi(t), Mpsi+СКО(t) и Mpsi-СКО(t) по ансамблю')  plt.ylabel('Mpsi, усл.ед')  plt.xlabel('t, усл.ед')  plt.grid(True)  plt.legend(['Mpsi', 'Mpsi+СКО', 'Mpsi-СКО'], loc=1)  fig\_snr.savefig('Pic\_block3\_MY\_ans\_sigma\_sigma\_const.png')   *# формируем список для энергии* list\_for\_energy\_u\_sigma\_const = find\_energy\_u(list\_for\_all\_u\_sigma\_const)  print('energy\_u\_sigma\_const', list\_for\_energy\_u\_sigma\_const)   *# Рисуем график энергии* fig\_snr = plt.figure()  plt.plot(list\_for\_T, list\_for\_energy\_u\_sigma\_const, color="red")  plt.title('Зависимость Eu(t)')  plt.ylabel('Eu, усл.ед')  plt.xlabel('t, усл.ед')  plt.grid(True)  plt.legend(['Eu'], loc=1)  fig\_snr.savefig('Pic\_block3\_E\_u\_sigma\_const.png')  plt.close()   *# # формируем список для энергии по ансамблю  # list\_for\_energy\_u\_ans\_sigma\_const = []  # for i in range (list\_for\_all\_u\_sigma\_const[0]):  # sum\_an = 0  # for j in range(len(list\_for\_all\_u\_sigma\_const)):  # sum\_an += list\_for\_all\_u\_sigma\_const[j][i]\*\*2  # list\_for\_energy\_u\_ans\_sigma\_const.append(sum\_an/len(list\_for\_all\_u\_sigma\_const))  # print('energy\_u\_ans\_sigma\_const', list\_for\_energy\_u\_ans\_sigma\_const)  #  # # Рисуем график энергии  # fig\_snr = plt.figure()  # plt.plot(list\_for\_T\_ans, list\_for\_energy\_u\_ans\_sigma\_const, color="red")  # plt.title('Зависимость Eu\_ans\_sigma\_const')  # plt.ylabel('Eu, усл.ед')  # plt.xlabel('ans, усл.ед')  # plt.grid(True)  # plt.legend(['Eu\_sigma\_const'], loc=1)  # fig\_snr.savefig('Pic\_block3\_E\_u\_sigma\_const.png')  # plt.close()   # Выводим таблицу* print(  ' \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_')  print(  '| sigma | psi(11) | Mpsi | Eu | EuA |')  print(  '|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|')  print('| ', round(list\_for\_sigma[0], 2), ' | ', round(list\_for\_all\_Y[0][99], 2),  ' | ', round(list\_for\_M\_Y[0], 2), ' | ', round(list\_for\_energy\_u[0], 2),  ' | ', round(list\_for\_energy\_u\_2[0], 2))  print(  '|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|')  print('| ', round(list\_for\_sigma[1], 2), ' | ', round(list\_for\_all\_Y[1][99], 2),  ' | ', round(list\_for\_M\_Y[1], 2), ' | ', round(list\_for\_energy\_u[1], 2),  ' | ', round(list\_for\_energy\_u\_2[1], 2))  print(  '|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|')  print('| ', round(list\_for\_sigma[2], 2), ' | ', round(list\_for\_all\_Y[2][99], 2),  ' | ', round(list\_for\_M\_Y[2], 2), ' | ', round(list\_for\_energy\_u[2], 2),  ' | ', round(list\_for\_energy\_u\_2[2], 2))  print(  '|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|')  print('| ', round(list\_for\_sigma[3], 2), ' | ', round(list\_for\_all\_Y[3][99], 2),  ' | ', round(list\_for\_M\_Y[3], 2), ' | ', round(list\_for\_energy\_u[3], 2),  ' | ', round(list\_for\_energy\_u\_2[3], 2))  print(  '|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|')  print('| ', round(list\_for\_sigma[4], 2), ' | ', round(list\_for\_all\_Y[4][99], 2),  ' | ', round(list\_for\_M\_Y[4], 2), ' | ', round(list\_for\_energy\_u[4], 2),  ' | ', round(list\_for\_energy\_u\_2[4], 2))  print(  '|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|')  print('| ', round(list\_for\_sigma[5], 2), ' | ', round(list\_for\_all\_Y[5][99], 2),  ' | ', round(list\_for\_M\_Y[5], 2), ' | ', round(list\_for\_energy\_u[5], 2),  ' | ', round(list\_for\_energy\_u\_2[5], 2))  print(  '|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|')  print('| ', round(list\_for\_sigma[6], 2), ' | ', round(list\_for\_all\_Y[6][99], 2),  ' | ', round(list\_for\_M\_Y[6], 2), ' | ', round(list\_for\_energy\_u[6], 2),  ' | ', round(list\_for\_energy\_u\_2[6], 2))  print(  '|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|')  print('| ', round(list\_for\_sigma[7], 2), ' | ', round(list\_for\_all\_Y[7][99], 2),  ' | ', round(list\_for\_M\_Y[7], 2), ' | ', round(list\_for\_energy\_u[7], 2),  ' | ', round(list\_for\_energy\_u\_2[7], 2))  print(  '|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|')  print('| ', round(list\_for\_sigma[8], 2), ' | ', round(list\_for\_all\_Y[8][99], 2),  ' | ', round(list\_for\_M\_Y[8], 2), ' | ', round(list\_for\_energy\_u[8], 2),  ' | ', round(list\_for\_energy\_u\_2[8], 2))  print(  '|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|')  print('| ', round(list\_for\_sigma[9], 2), ' | ', round(list\_for\_all\_Y[9][99], 2),  ' | ', round(list\_for\_M\_Y[9], 2), ' | ', round(list\_for\_energy\_u[9], 2),  ' | ', round(list\_for\_energy\_u\_2[9], 2))  print(  '|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|')  print('| ', round(list\_for\_sigma[10], 2), ' | ', round(list\_for\_all\_Y[10][99], 2),  ' | ', round(list\_for\_M\_Y[10], 2), ' | ', round(list\_for\_energy\_u[10], 2),  ' | ', round(list\_for\_energy\_u\_2[10], 2))  print(  '|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|')  print('len\_list\_psi', len(list\_for\_all\_Y[0]))  *# Запись данных в файл* f = open('table.txt', 'w')  f.write(  ' \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n')  f.write(  '| sigma | psi(11) | MY | Eu | EuA |\n')  f.write(  '|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\n')  f.write('| ')  f.write(str(round(list\_for\_sigma[0], 2)))  f.write(' | ')  f.write(str(round(list\_for\_all\_Y[0][99], 2)))  f.write(' | ')  f.write(str(round(list\_for\_M\_Y[0], 2)))  f.write(' | ')  f.write(str(round(list\_for\_energy\_u[0], 2)))  f.write(' | ')  f.write(str(round(list\_for\_energy\_u\_2[0], 2)))  f.write(  '\n|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\n')  f.write('| ')  f.write(str(round(list\_for\_sigma[1], 2)))  f.write(' | ')  f.write(str(round(list\_for\_all\_Y[1][99], 2)))  f.write(' | ')  f.write(str(round(list\_for\_M\_Y[1], 2)))  f.write(' | ')  f.write(str(round(list\_for\_energy\_u[1], 2)))  f.write(' | ')  f.write(str(round(list\_for\_energy\_u\_2[1], 2)))  f.write(  '\n|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\n')  f.write('| ')  f.write(str(round(list\_for\_sigma[2], 2)))  f.write(' | ')  f.write(str(round(list\_for\_all\_Y[2][99], 2)))  f.write(' | ')  f.write(str(round(list\_for\_M\_Y[2], 2)))  f.write(' | ')  f.write(str(round(list\_for\_energy\_u[2], 2)))  f.write(' | ')  f.write(str(round(list\_for\_energy\_u\_2[2], 2)))  f.write(  '\n|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\n')  f.write('| ')  f.write(str(round(list\_for\_sigma[3], 2)))  f.write(' | ')  f.write(str(round(list\_for\_all\_Y[3][99], 2)))  f.write(' | ')  f.write(str(round(list\_for\_M\_Y[3], 2)))  f.write(' | ')  f.write(str(round(list\_for\_energy\_u[3], 2)))  f.write(' | ')  f.write(str(round(list\_for\_energy\_u\_2[3], 2)))  f.write(  '\n|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\n')  f.write('| ')  f.write(str(round(list\_for\_sigma[4], 2)))  f.write(' | ')  f.write(str(round(list\_for\_all\_Y[4][99], 2)))  f.write(' | ')  f.write(str(round(list\_for\_M\_Y[4], 2)))  f.write(' | ')  f.write(str(round(list\_for\_energy\_u[4], 2)))  f.write(' | ')  f.write(str(round(list\_for\_energy\_u\_2[4], 2)))  f.write(  '\n|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\n')  f.write('| ')  f.write(str(round(list\_for\_sigma[5], 2)))  f.write(' | ')  f.write(str(round(list\_for\_all\_Y[5][99], 2)))  f.write(' | ')  f.write(str(round(list\_for\_M\_Y[5], 2)))  f.write(' | ')  f.write(str(round(list\_for\_energy\_u[5], 2)))  f.write(' | ')  f.write(str(round(list\_for\_energy\_u\_2[5], 2)))  f.write(  '\n|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\n')  f.write('| ')  f.write(str(round(list\_for\_sigma[6], 2)))  f.write(' | ')  f.write(str(round(list\_for\_all\_Y[6][99], 2)))  f.write(' | ')  f.write(str(round(list\_for\_M\_Y[6], 2)))  f.write(' | ')  f.write(str(round(list\_for\_energy\_u[6], 2)))  f.write(' | ')  f.write(str(round(list\_for\_energy\_u\_2[6], 2)))  f.write(  '\n|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\n')  f.write('| ')  f.write(str(round(list\_for\_sigma[7], 2)))  f.write(' | ')  f.write(str(round(list\_for\_all\_Y[7][99], 2)))  f.write(' | ')  f.write(str(round(list\_for\_M\_Y[7], 2)))  f.write(' | ')  f.write(str(round(list\_for\_energy\_u[7], 2)))  f.write(' | ')  f.write(str(round(list\_for\_energy\_u\_2[7], 2)))  f.write(  '\n|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\n')  f.write('| ')  f.write(str(round(list\_for\_sigma[8], 2)))  f.write(' | ')  f.write(str(round(list\_for\_all\_Y[8][99], 2)))  f.write(' | ')  f.write(str(round(list\_for\_M\_Y[8], 2)))  f.write(' | ')  f.write(str(round(list\_for\_energy\_u[8], 2)))  f.write(' | ')  f.write(str(round(list\_for\_energy\_u\_2[8], 2)))  f.write(  '\n|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\n')  f.write('| ')  f.write(str(round(list\_for\_sigma[9], 2)))  f.write(' | ')  f.write(str(round(list\_for\_all\_Y[9][99], 2)))  f.write(' | ')  f.write(str(round(list\_for\_M\_Y[9], 2)))  f.write(' | ')  f.write(str(round(list\_for\_energy\_u[9], 2)))  f.write(' | ')  f.write(str(round(list\_for\_energy\_u\_2[9], 2)))  f.write(  '\n|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\n')  f.write('| ')  f.write(str(round(list\_for\_sigma[10], 2)))  f.write(' | ')  f.write(str(round(list\_for\_all\_Y[10][99], 2)))  f.write(' | ')  f.write(str(round(list\_for\_M\_Y[10], 2)))  f.write(' | ')  f.write(str(round(list\_for\_energy\_u[10], 2)))  f.write(' | ')  f.write(str(round(list\_for\_energy\_u\_2[10], 2)))  f.write(  '\n|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\n')  f.close()   *# \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ БЛОК 4 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  # Создадим списк для списков данных x, s, u, Y, sigma* list\_for\_all\_x\_omega = []  list\_for\_all\_s\_omega = []  list\_for\_all\_u\_omega = []  list\_for\_all\_Y\_omega = []   list\_for\_omega = [0.01, 0.6, 0.99]  for n in range(3):  *# значение шума в 0 момент времени* e = np.random.normal(0, float(list\_for\_sigma[5]), 1)  *# формируем список значений шума* list\_for\_e = [float(e)]  *# формируем список значений кол-ва товаров на складе* list\_for\_x = [float(x)]  *# формируем список значений величины спроса* list\_for\_s = [float(s)]  *# формируем список значений объема поставок* list\_for\_u = [float(u)]  *# формируем список значений макропеременной* list\_for\_Y = [float(Y)]  for t in range(1, 100):  *# получаем значение спроса в момент времени t и добавляем его в список* s\_t = a \* list\_for\_s[t - 1] + b  list\_for\_s.append(float(s\_t))  *# получаем новое значение шума и добавляем его в список* e\_t = np.random.normal(0, float(list\_for\_sigma[5]), 1)  list\_for\_e.append(float(e\_t))  *# получаем новое значение кол-ва товаров на складе в момент времени t и добавляем его в список* x\_t = list\_for\_x[t - 1] + list\_for\_u[t - 1] + c \* list\_for\_e[t - 1] + e\_t - s\_t  list\_for\_x.append(float(x\_t))  *# получем значение макропеременной и добавляем его в список* Y\_t = x\_t - x\_  list\_for\_Y.append(float(Y\_t))  *# получаем момент спроса в t+1* s\_t\_plus\_1 = a \* s\_t + b  *# получим значение управления по методу акар* u\_a = (list\_for\_omega[n] + c) \* (-Y\_t) + x\_ + c \* list\_for\_omega[n] \* list\_for\_Y[t - 1] + s\_t\_plus\_1 - x\_t   if u\_a < Umin:  *# получаем объём поставок в момент времени t* u\_t = 0  list\_for\_u.append(float(u\_t))  elif u\_a < Umax:  u\_t = u\_a  list\_for\_u.append(float(u\_t))  else:  u\_t = Umax  list\_for\_u.append(float(u\_t))  *# добавляем полученные данные в массив с массивами* list\_for\_all\_x\_omega.append(list\_for\_x)  list\_for\_all\_s\_omega.append(list\_for\_s)  list\_for\_all\_u\_omega.append(list\_for\_u)  list\_for\_all\_Y\_omega.append(list\_for\_Y)  *# # Выводим зачения на экран  # print('x: ', list\_for\_x)  # print('s: ', list\_for\_s)  # print('u: ', list\_for\_u)  # print('Y: ', list\_for\_Y)   # #Рисуем графики* list\_for\_t\_omega = []  for num in range(len(list\_for\_all\_x\_omega[0])):  list\_for\_t\_omega.append(num)   fig = plt.figure()  plt.plot(list\_for\_t\_omega, list\_for\_all\_x\_omega[0], 'm', list\_for\_t\_omega, list\_for\_all\_x\_omega[1], 'k',  list\_for\_t\_omega, list\_for\_all\_x\_omega[2], 'y')  plt.title('Зависимость x(t) при разном значении omega')  plt.ylabel('x, усл.ед')  plt.xlabel('t, усл.ед')  plt.grid(True)  plt.legend(['x при omega = 0,01', 'x при omega = 0,6', 'x при omega = 0,99'], loc=1)  fig.savefig('Pic\_block4\_for\_x\_t\_omega.png')  plt.close()  fig\_1 = plt.figure()  plt.plot(list\_for\_t\_omega, list\_for\_all\_s\_omega[0], 'm', list\_for\_t\_omega, list\_for\_all\_s\_omega[1], 'k',  list\_for\_t\_omega, list\_for\_all\_s\_omega[2], 'y')  plt.title('Зависимость s(t) при разном значении omega')  plt.ylabel('s, усл.ед')  plt.xlabel('t, усл.ед')  plt.grid(True)  plt.legend(['s при omega = 0,01', 's при omega = 0,6', 's при omega = 0,99'], loc=1)  *# plt.show()* fig\_1.savefig('Pic\_block4\_for\_s\_t\_omega.png')  plt.close()  fig\_2 = plt.figure()  plt.plot(list\_for\_t\_omega, list\_for\_all\_u\_omega[0], 'm', list\_for\_t\_omega, list\_for\_all\_u\_omega[1], 'k',  list\_for\_t\_omega, list\_for\_all\_u\_omega[2], 'y')  plt.title('Зависимость u(t) при разном значении omega')  plt.ylabel('u, усл.ед')  plt.xlabel('t, усл.ед')  plt.grid(True)  plt.legend(['u при omega = 0,01', 'u при omega = 0,6', 'u при omega = 0,99'], loc=1)  *# plt.show()* fig\_2.savefig('Pic\_block4\_for\_u\_t\_omega.png')  plt.close()  fig\_2 = plt.figure()  plt.plot(list\_for\_t\_omega, list\_for\_all\_Y\_omega[0], 'm', list\_for\_t\_omega, list\_for\_all\_Y\_omega[1], 'k',  list\_for\_t\_omega, list\_for\_all\_Y\_omega[2], 'y')  plt.title('Зависимость psi(t) при разном значении omega')  plt.ylabel('psi, усл.ед')  plt.xlabel('t, усл.ед')  plt.grid(True)  plt.legend(['psi при omega = 0,01', 'psi при omega = 0,6', 'psi при omega = 0,99'], loc=1)  *# plt.show()* fig\_2.savefig('Pic\_block4\_for\_Y\_t\_omega.png')  plt.close()   *# \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ БЛОК 5 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  # Создадим списк для списков данных x, s, u, Y, sigma* list\_for\_all\_x\_nach\_zn = []  list\_for\_all\_s\_nach\_zn = []  list\_for\_all\_u\_nach\_zn = []  list\_for\_all\_Y\_nach\_zn = []   list\_for\_xo = [100, 160, 250]  list\_for\_so = [1, 5, 10]  *# list\_for\_uo = [0.01, 0.6, 0.99]* for n in range(3):  *# значение шума в 0 момент времени* e = np.random.normal(0, float(list\_for\_sigma[5]), 1)  *# формируем список значений шума* list\_for\_e = [float(e)]  *# формируем список значений кол-ва товаров на складе* list\_for\_x = [float(list\_for\_xo[n])]  *# формируем список значений величины спроса* list\_for\_s = [float(list\_for\_so[n])]  *# формируем список значений объема поставок* list\_for\_u = [float(u)]  *# формируем список значений макропеременной* list\_for\_Y = [float(Y)]  for t in range(1, 100):  *# получаем значение спроса в момент времени t и добавляем его в список* s\_t = a \* list\_for\_s[t - 1] + b  list\_for\_s.append(float(s\_t))  *# получаем новое значение шума и добавляем его в список* e\_t = np.random.normal(0, float(list\_for\_sigma[5]), 1)  list\_for\_e.append(float(e\_t))  *# получаем новое значение кол-ва товаров на складе в момент времени t и добавляем его в список* x\_t = list\_for\_x[t - 1] + list\_for\_u[t - 1] + c \* list\_for\_e[t - 1] + e\_t - s\_t  list\_for\_x.append(float(x\_t))  *# получем значение макропеременной и добавляем его в список* Y\_t = x\_t - x\_  list\_for\_Y.append(float(Y\_t))  *# получаем момент спроса в t+1* s\_t\_plus\_1 = a \* s\_t + b  *# получим значение управления по методу акар* u\_a = (w + c) \* (-Y\_t) + x\_ + c \* w \* list\_for\_Y[t - 1] + s\_t\_plus\_1 - x\_t   if u\_a < Umin:  *# получаем объём поставок в момент времени t* u\_t = 0  list\_for\_u.append(float(u\_t))  elif u\_a < Umax:  u\_t = u\_a  list\_for\_u.append(float(u\_t))  else:  u\_t = Umax  list\_for\_u.append(float(u\_t))  *# добавляем полученные данные в массив с массивами* list\_for\_all\_x\_nach\_zn.append(list\_for\_x)  list\_for\_all\_s\_nach\_zn.append(list\_for\_s)  list\_for\_all\_u\_nach\_zn.append(list\_for\_u)  list\_for\_all\_Y\_nach\_zn.append(list\_for\_Y)  *# # Выводим зачения на экран  # print('x: ', list\_for\_x)  # print('s: ', list\_for\_s)  # print('u: ', list\_for\_u)  # print('Y: ', list\_for\_Y)   # #Рисуем графики* list\_for\_t\_omega = []  for num in range(len(list\_for\_all\_x\_nach\_zn[0])):  list\_for\_t\_omega.append(num)   fig = plt.figure()  plt.plot(list\_for\_t\_omega, list\_for\_all\_x\_nach\_zn[0], 'm', list\_for\_t\_omega, list\_for\_all\_x\_nach\_zn[1], 'k',  list\_for\_t\_omega, list\_for\_all\_x\_nach\_zn[2], 'y')  plt.title('Зависимость x(t) при разных начальных значениях')  plt.ylabel('x')  plt.xlabel('t')  plt.grid(True)  plt.legend(['x при x0=100, s0=1', 'x при x0=160, s0=5', 'x при x0=250, s0=10'], loc=1)  fig.savefig('Pic\_block5\_for\_x\_t\_nach\_zn.png')  plt.close()  fig\_1 = plt.figure()  plt.plot(list\_for\_t\_omega, list\_for\_all\_s\_nach\_zn[0], 'm', list\_for\_t\_omega, list\_for\_all\_s\_nach\_zn[1], 'k',  list\_for\_t\_omega, list\_for\_all\_s\_nach\_zn[2], 'y')  plt.title('Зависимость s(t) при разных начальных значениях')  plt.ylabel('s')  plt.xlabel('t')  plt.grid(True)  plt.legend(['s при x0=100, s0=1', 's при x0=160, s0=5', 's при x0=250, s0=10'], loc=1)  *# plt.show()* fig\_1.savefig('Pic\_block5\_for\_s\_t\_nach\_zn.png')  plt.close()  fig\_2 = plt.figure()  plt.plot(list\_for\_t\_omega, list\_for\_all\_u\_nach\_zn[0], 'm', list\_for\_t\_omega, list\_for\_all\_u\_nach\_zn[1], 'k',  list\_for\_t\_omega, list\_for\_all\_u\_nach\_zn[2], 'y')  plt.title('Зависимость u(t) при разных начальных значениях')  plt.ylabel('u')  plt.xlabel('t')  plt.grid(True)  plt.legend(['u при x0=100, s0=1', 'u при x0=160, s0=5', 'u при x0=250, s0=10'], loc=1)  *# plt.show()* fig\_2.savefig('Pic\_block5\_for\_u\_t\_nach\_zn.png')  plt.close()  fig\_2 = plt.figure()  plt.plot(list\_for\_t\_omega, list\_for\_all\_Y\_nach\_zn[0], 'm', list\_for\_t\_omega, list\_for\_all\_Y\_nach\_zn[1], 'k',  list\_for\_t\_omega, list\_for\_all\_Y\_nach\_zn[2], 'y')  plt.title('Зависимость psi(t) при разных начальных значениях')  plt.ylabel('psi')  plt.xlabel('t')  plt.grid(True)  plt.legend(['psi при x0=100, s0=1', 'psi при x0=160, s0=5', 'psi при x0=250, s0=10'], loc=1)  *# plt.show()* fig\_2.savefig('Pic\_block5\_for\_Y\_t\_nach\_zn.png')  plt.close()       *#Присваиваем значения введенные пользователем переменным* def on\_click():  a = form.for\_a.text()  b = form.for\_b.text()  c = form.for\_c.text()  w = form.for\_w.text()  x\_ = form.for\_x\_.text()  Umin = form.for\_Umin.text()  Umax = form.for\_Umax.text()  s = form.for\_s0.text()  x = form.for\_x0.text()  u = form.for\_u0.text()  sigma\_const = form.for\_sigma.text()  print\_perem(float(a), float(b), float(c), float(w), float(x\_), float(Umin), float(Umax), float(s), float(x), float(u), float(sigma\_const))  QMessageBox.information(window, "Вычисления окончены!", "Проверьте папку", QMessageBox.Ok)         if \_\_name\_\_ == '\_\_main\_\_':   print(os.path.realpath(\_\_file\_\_))  dirname, filename = os.path.split(os.path.realpath(\_\_file\_\_))  print(dirname)  Form, Window = uic.loadUiType(dirname + "\\diplom.ui")   app = QApplication(sys.argv)  window = Window()  form = Form()  form.setupUi(window)  window.show()   form.btn\_count.clicked.connect(on\_click)     sys.exit(app.exec\_()) |