

МИНИСТЕРСТВО ВЫСШЕГО ОБРАЗОВАНИЯ, НАУКИ И ИННОВАЦИЙ
РЕСПУБЛИКИ УЗБЕКИСТАН

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ТАШКЕНТСКИЙ ИНСТИТУТ ИНЖЕНЕРОВ ИРРИГАЦИИ И
МЕХАНИЗАЦИИ СЕЛЬСКОГО ХОЗЯЙСТВА»

КАБИЛЬДЖАНОВ АЛЕКСАНДР САБИТОВИЧ

ПРОГРАММИРОВАНИЕ

Основы работы в MATLAB
Учебник



Ташкент-2025

**МИНИСТЕРСТВО ВЫСШЕГО ОБРАЗОВАНИЯ, НАУКИ И
ИННОВАЦИЙ РЕСПУБЛИКИ УЗБЕКИСТАН**

**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ТАШКЕНТСКИЙ ИНСТИТУТ ИНЖЕНЕРОВ ИРРИГАЦИИ И
МЕХАНИЗАЦИИ СЕЛЬСКОГО ХОЗЯЙСТВА»**

КАБИЛЬДЖАНОВ АЛЕКСАНДР САБИТОВИЧ

ПРОГРАММИРОВАНИЕ

Основы работы в MATLAB

Учебник

Ташкент-2025

УДК: 004.42

Кабильджанов А.С. – Программирование. Основы работы в MATLAB.

Учебник. - Ташкент. «Фан». 2025. -217 с.

Учебник предназначен для студентов, обучающихся по направлению бакалавриата **5311000** - «Автоматизация и управление технологическими процессами и производством» (в водном хозяйстве). Учебник отражает основные вопросы программирования в MATLAB.

Учебник призван закрепить теоретические знания, полученные студентами при изучении дисциплины «Программирование». Приведенные в учебнике примеры и задания позволяют получить практические навыки программирования в MATLAB и моделирования динамических систем в SIMULINK.

Рецензенты:

Назаров А.И. - доцент, к.т.н.

Мухамедиева Д.Т. - профессор, д.т.н.

©Национальный исследовательский университет «Ташкентский институт инженеров ирригации и механизации сельского хозяйства» 2025 г.

ОГЛАВЛЕНИЕ

№	Наименование	Стр.
	Введение	6
	Часть I. Основы программирования	8
	Глава 1. Алгоритмизация	8
§1.1.	Блок-схемы алгоритмов	8
§1.2.	Основные алгоритмические конструкции	9
§1.3.	Типы блок-схем	13
	Контрольные вопросы	13
	Глава 2. Общие сведения о среде MATLAB	14
§2.1.	Общие сведения о MATLAB	14
§2.2.	Пакеты расширения MATLAB	17
§2.3.	Интерфейс системы MATLAB	29
	Контрольные вопросы	34
	Глава 3. Основные выражения и операции	34
§3.1.	Математические выражения	34
§3.2.	Арифметические операции с простыми переменными	36
§3.3.	Логические операторы	37
§3.4.	Основные математические функции	37
	Контрольные вопросы	38
	Глава 4. Операции над матрицами и векторами	38
§4.1.	Вектора и матрицы в MATLAB	38
§4.2.	Операции над матрицами и векторами	41
	Контрольные вопросы	48
	Глава 5. Организация ветвления и циклов	50
§5.1.	Условные операторы	50
§5.2.	Циклы в MATLAB	52
	Контрольные вопросы	53
	Глава 6. Построение графиков и диаграмм	54

§6.1. Двумерная графика в MATLAB	54
§6.2. Трёхмерная графика в MATLAB	57
§6.3. Построение диаграмм и гистограмм в MATLAB	59
§6.4. Управление световыми эффектами в MATLAB	61
Контрольные вопросы	62
Глава 7. Программирование процедур и функций, работа с файлами	63
§7.1. Программирование процедур и функций	63
§7.2. Работа с файлами в MATLAB	64
§7.3. Специальные функции для работы с передаточными функциями.....	65
Контрольные вопросы	74
Часть II. Основы работы с данными и моделями.....	75
Глава 8. Функции извлечения данных из модели и анализа свойств модели	75
§8.1. Функции извлечения данных из модели	75
§8.2. Функции анализа свойств модели	77
Контрольные вопросы	79
Глава 9. LTViewer - интерактивный обозреватель свойств линейных моделей	80
§9.1. Задание идентификаторов созданных LTI - систем во входных параметрах	80
§9.2. Представление LTI-системы в пространстве состояний	81
Контрольные вопросы	83
Часть III. Пакет расширения SIMULINK	84
Глава 10. Общие сведения о программном комплексе SIMULINK	84
§10.1. Основные возможности программного комплекса	84
§10.2. Создание нового окна модели	85
§10.3. Боковая панель инструментов	88

§10.4. Описание библиотеки SIMULINK	89
Контрольные вопросы	92
Глава 11. Создание и анализ моделей САУ в программе SIMULINK..	93
§11.1. Создание, настройка, сохранение и вывод на печать модели	93
§11.2. Линеаризация и анализ моделей инструментом Linear analysis	105
Контрольные вопросы	112
Часть IV. Описания практических занятий	113
Практическое занятие №1. Простейшие вычисления в MATLAB.....	113
Практическое занятие №2. Работа с вектор-столбцами и вектор-строками	114
Практическое занятие №3. Операции над массивами и матрицами	118
Практическое занятие №4. Работа с двумерными массивами и матрицами	122
Практическое занятие №5. Построение и работа с блочными матрицами	124
Практическое занятие №6. Поэлементные операции над матрицами	129
Практическое занятие №7. Операторы цикла в MATLAB	131
Практическое занятие №8. Условные операторы	134
Практическое занятие №9. Вывод графиков, диаграмм и гистограмм ..	135
Практическое занятие №10. Работа с передаточными функциями	140
Практическое занятие №11. Работа с функциями извлечения данных из модели и анализа свойств модели	144
Практическое занятие №12. Интерфейс браузера библиотек SIMULINK	148
Практическое занятие №13. Создание моделей в SIMULINK	153
Практическое занятие №14. Анализ моделей в SIMULINK	156
Список литературы	161
Варианты для самостоятельного выполнения	164

Введение

На сегодняшнее время создано и успешно применяется на практике множество математических пакетов, обладающих мощными символьными процессорами и предназначенных для решения широкого круга научных и инженерных задач. Среди наиболее популярных ведущее место занимают: MATLAB, Mathematica, Maple, MathCAD, Statistica и другие. Применительно к задачам синтеза и анализа систем автоматического управления (САУ) вполне достаточно нескольких программ, среди которых ведущее место занимает MATLAB.

Целью учебника является систематизация материала по основам программирования и работы в математическом пакете MATLAB, созданию структурных моделей и анализу систем в пакете SIMULINK.

Материал, изложенный в учебнике, способствует закреплению теоретических знаний и практических навыков по программированию и работе в среде MATLAB, что в определенной степени является базой при изучении таких специальных дисциплин, как «Компьютерное моделирование систем управления», «Теория автоматического управления» и «АСУТП водного хозяйства». Учебник предназначен для студентов специальности 5311000 - «Автоматизация и управление технологическими процессами и производством» (в водном хозяйстве) и может быть использован применительно к другим аналогичным специальностям.

Учебник состоит из четырех частей. Основой для изучения курса «Программирование» служит пакет MATLAB и его расширения. В первой части, состоящей из семи глав, изложены вопросы разработки алгоритмов для решения вычислительных задач, даны общие сведения о пакете MATLAB, приводятся описания пакетов расширения, интерфейса и основных конструкций языка программирования MATLAB. Подробно описаны операции над матрицами и векторами, организация разветвлений и циклов, графическая поддержка, а также программирование функций и работа с файлами в MATLAB. Вторая часть учебника состоит из двух глав и содержит материал по основам

работы с данными и анализу моделей в MATLAB, а также описание LTiviewer - интерактивного обозревателя свойств линейных моделей. В третьей части, состоящей из двух глав, приводятся основные сведения о пакете расширения SIMULINK и браузере библиотеки SIMULINK. Подробно рассмотрены вопросы создания и анализа моделей САУ в программе SIMULINK. Четвертая часть содержит описания четырнадцати практических занятий и варианты индивидуальных заданий.

Материал учебника изложен в порядке возрастания сложности и в рамках тематических разделов сложные задачи реструктурированы на простые, которые составляют содержание примеров. Причем, простые задачи иллюстрируют материал первых трех частей, а более сложные задачи приведены в четвертой части учебника.

ЧАСТЬ I. ОСНОВЫ ПРОГРАММИРОВАНИЯ

ГЛАВА 1. АЛГОРИТМИЗАЦИЯ

§1.1. Блок-схемы алгоритмов

При разработке программы автоматизации какого-либо технологического процесса с выполнением ее на компьютере проходят три этапа: *формализация, алгоритмизация и программирование*.

Формализации предшествует тщательное изучение объекта автоматизации и постановка задачи. Далее осуществляется детальная проработка и формализация задачи управления, в ходе которых строится математическая модель объекта управления, выбираются методы реализации математической модели и оптимального управления. Уровень проработки и формализации задачи управления должен быть таким, чтобы на его основе можно было легко разработать алгоритмы и написать программы.

При алгоритмизации детализируется метод решения задачи и формально описывается последовательность выполняемых операций, которые приводят к решению задачи. Формальное описание задачи, как правило, оформляется в виде *блок-схемы алгоритма*, или, коротко, *блок-схемы*.

Блок-схема, являясь одним из способов описания алгоритма, производит описание структуры алгоритма при помощи набора блоков, оформленных в виде геометрических фигур. Связи между блоками задаются линиями, которые определяют порядок выполнения отдельных команд. Такой способ имеет ряд преимуществ: наглядности, обеспечивающую «удобочитаемость» алгоритма; явное задание порядка выполнения отдельных команд. Как уже отмечалось выше, в блок-схеме каждому отдельному блоку соответствует определенная геометрическая фигура или совокупность фигур, связанных линиями.

Основные блоки, которые используются при построении блок-схем алгоритмов, приведены на рис. 1.1.

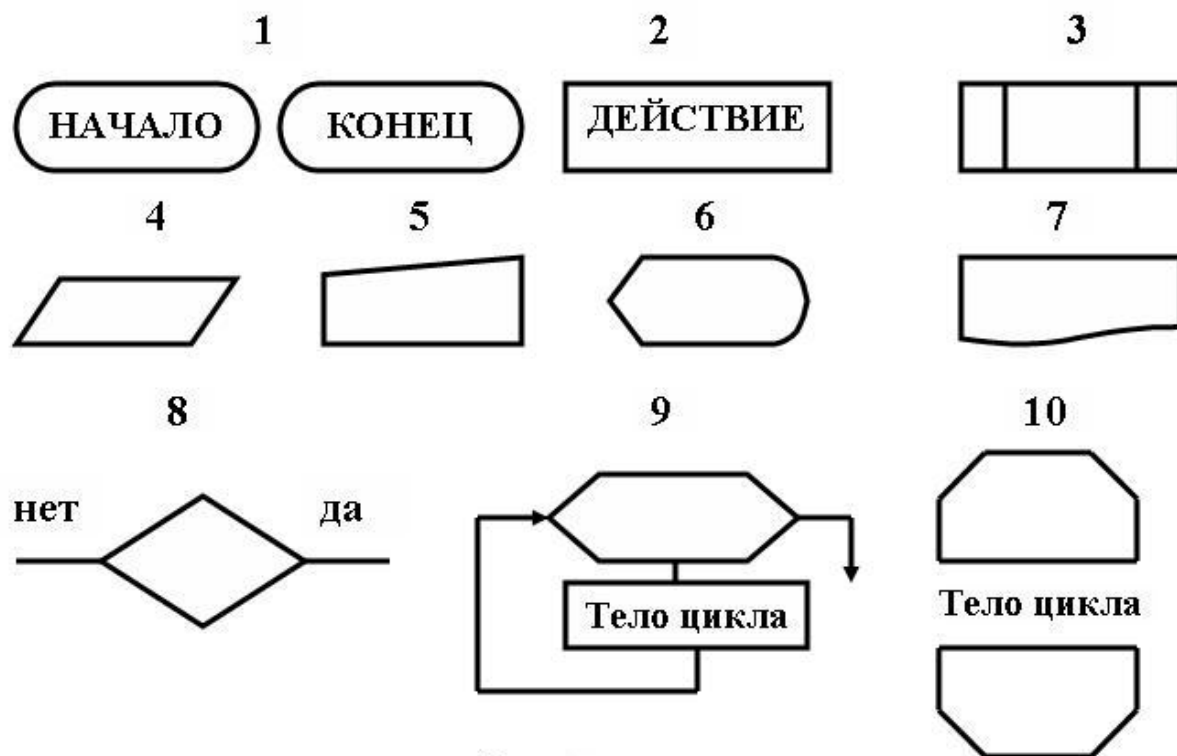


Рис. 1.1. Основные блоки для построения блок-схем алгоритмов

(1) блок начала/конца алгоритма; (2) операторный блок - предназначен для задания отдельных вычислительных действий; (3) блок подпрограммы- предназначен для задания *предопределенного процесса*; (4) блок *ввода/вывода данных с неопределенного носителя*; (5) блок ввода информации с клавиатуры; (6) блок вывода информации на монитор; (7) блок вывода информации на печатающее устройство; (8) блок проверки логического условия; (9) блок для организации циклических вычислений; (10) два блока для задания границ цикла: начало цикла, конец цикла

§1.2. Основные алгоритмические конструкции

Среди различных последовательностей выполнения элементарных действий алгоритма можно выделить три вида алгоритмических конструкций: *линейные, разветвляющиеся, циклические и рекурсивные*.

Алгоритмическую конструкцию называют *линейной*, если она реализуется в виде последовательности шагов, причем в каждый шаг алгоритма выполняется только один раз. После каждого i -го шага выполняется $(i+1)$ -ый шаг, если i -ый шаг не является концом алгоритма. Пример линейной конструкции алгоритма приведен на рис. 1.2.

Алгоритмическая конструкция называется *разветвляющейся*, если в ней в зависимости от значения входных данных существует выбор между двумя альтернативами. Применительно к конкретному набору входных дан-

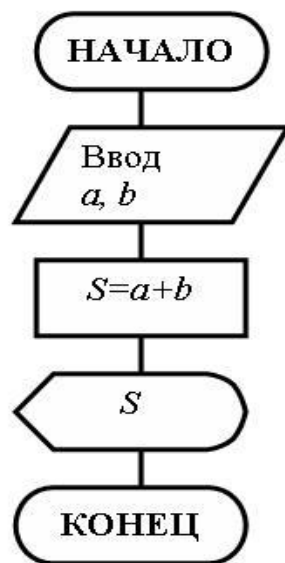


Рис. 1.2. Пример линейной конструкции алгоритма

ных разветвляющийся алгоритм имеет вид линейного. При этом существует полное и неполное ветвления. Полное ветвление в алгоритме содержит две ветви (рис. 1.3, а), в каждой из которых выполняется определенная последовательность действий. Затем ветви сливаются в общей точке, что обеспечивает продолжение выполнения алгоритма независимо от выбранного пути. При неполном ветвлении (рис. 1.3, б) производится выполнение некоторой последовательности действий

алгоритма только на одной ветви. Второй ветви, соответствующей одному из результатов проверки, нет. Управление после окончания проверки переходит к точке слияния. При этом никакие действия не выполняются.

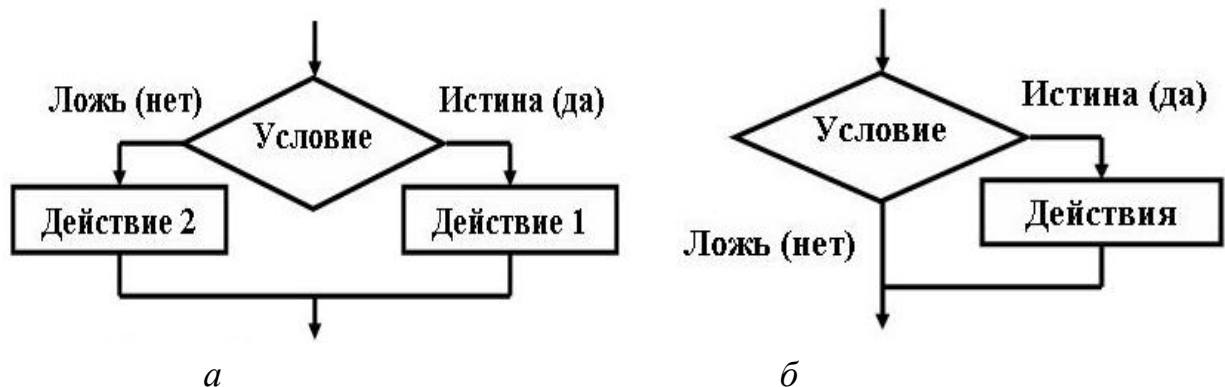


Рис. 1.3. Примеры разветвляющейся конструкции алгоритма
а- полное ветвление; б- неполное ветвление

Алгоритмическую конструкцию называют *циклической*, если в зависимости от входных данных или условий задачи, какая-то группа последовательных действий алгоритма выполняется многократно. Группа действий, повторяющихся на каждом шаге цикла, называется телом цикла. Особенностью

любой циклической конструкции является наличие элементов ветвящейся конструкции.

На практике выделяют три типа циклических алгоритмов: *арифметический цикл* или *цикл с параметром*, *цикл с предусловием*, *итерационный цикл* или *цикл с постусловием*.

Число шагов *арифметического цикла* определяется изменением параметра K , зависящего от начального значения N_n , конечного значения N_k и шага изменения h . На первом шаге цикла значение параметра равно N_n , затем каждый раз увеличивается на величину h : $N_n + h$, $N_n + 2h$ и т.д. Цикл заканчивается на n шаге, при котором выполняется условие $N_n + nh \leq N_k \& N_n + (n+1)h > N_k$. На рис. 1.4 приведен пример цикла с параметром.

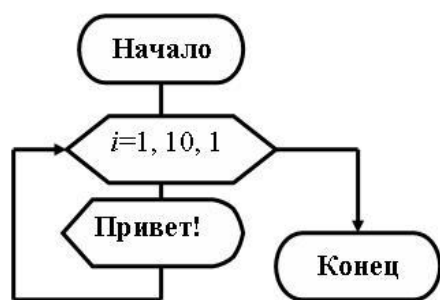


Рис. 1.4. Пример цикла с параметром

Количество шагов в *цикле с предусловием* заранее не известно и зависит от входных данных задачи. Перед выполнением очередного шага цикла сначала проверяется значение определенного условия. Тело цикла выполняется, если значение условия истинно. Затем вновь происходит передача управления проверке условия. Последовательность действий повторяется циклически до тех пор, пока условие не примет значение ложь, что приводит к завершению цикла. Если условие изначально ложно, тело цикла не выполнится ни разу. Блок-схема цикла с предусловием представлена на рис. 1.5 и включает два способа. Первый способ использует условный блок, второй способ - блок границы цикла.

Последовательность действий повторяется циклически до тех пор, пока условие не примет значение ложь, что приводит к завершению цикла. Если условие изначально ложно, тело цикла не выполнится ни разу. Блок-схема цикла с предусловием представлена на рис. 1.5 и включает два способа. Первый способ использует условный блок, второй способ - блок границы цикла.

Цикл с постусловием отличается от цикла с предусловием тем, что его тело выполняется хотя бы один раз до проверки условия. Причем, тело цикла выполняется до тех пор, пока условие не примет значение ложь. Выполнение команд прекращается, как только значение условия становится истинным.

Блок-схема цикла с постусловием приведена на рис. 1.6 и включает два способа. Первый способ использует условный блок, второй способ - блок управления.

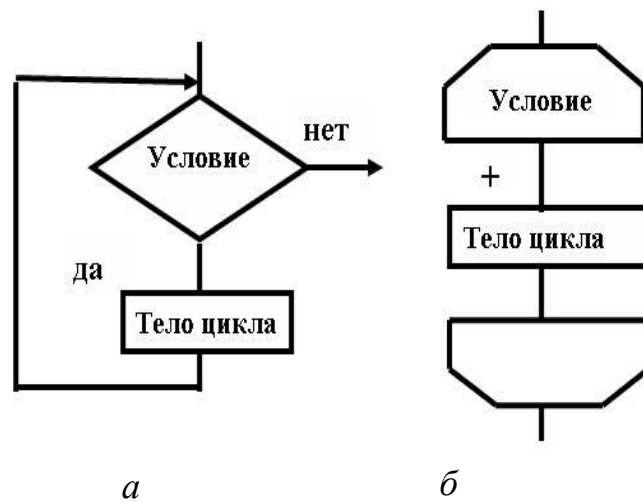


Рис. 1.5. Примеры цикла с предусловием
 а- при помощи условного блока;
 б- при помощи блока границы цикла

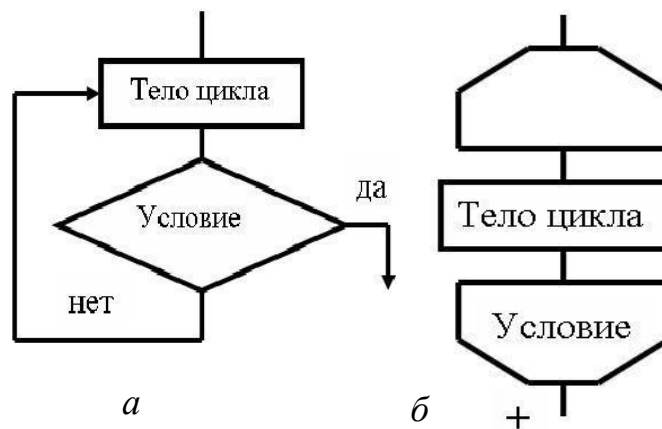


Рис. 1.6. Примеры цикла с постусловием
 а- при помощи условного блока;
 б- при помощи блока управления

Существуют также *рекурсивные алгоритмы*, в которых процесс выполнения команд организован таким образом, что на определенном шаге алгоритм обращается сам к себе прямо или косвенно.

§1.3. Типы блок-схем

На практике используются три типа блок-схем:

- *системная блок-схема* показывает, какие устройства используют для ввода, вывода и хранения данных. Алгоритм решения задачи изображается в виде единственного блока (рис. 1.7, а);
- *укрупненная блок-схема* детализируется до укрупненных блоков, где каждый содержит несколько команд программы (рис. 1.7, б);
- *подробные (детальные) блок-схемы*, в которых происходит дальнейшая детализация алгоритмов укрупненной блок-схемы так, что каждый блок представляет одну или две-три команды (рис. 1.7, в).

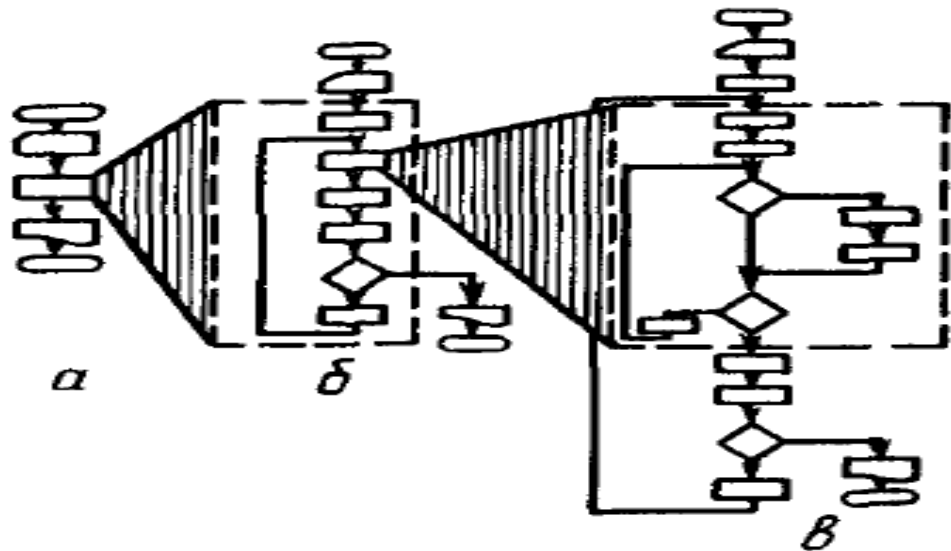


Рис. 1.7. Типы блок-схем

а - системная; б – укрупненная; в- детальная

Контрольные вопросы

1. Что такое алгоритм.
2. Назовите способы представления алгоритмов.
3. Какое формальное описание задачи называется блок-схемой?
4. Какие основные конструкции используются при построении блок-схем?
5. Какие алгоритмические конструкции используются на практике?
6. Опишите линейную алгоритмическую конструкцию.

7. Опишите разветвляющуюся алгоритмическую конструкцию.
8. Опишите циклическую алгоритмическую конструкцию.
9. Опишите рекурсивную алгоритмическую конструкцию.
10. Какие типы блок-схем используются на практике?
11. Перечислите этапы создания программного продукта.
12. Для чего выполняется структуризация программ?
13. Перечислите основные особенности алгоритмов.
14. Что такое блок-схема, каковы правила составления блок-схем?
15. Какие виды циклов вы знаете?
16. Объясните работу циклов со счетчиком.
17. В чем отличие циклов с предусловием от циклов с постусловием?
18. Что такое сложные циклы?

2. ГЛАВА. ОБЩИЕ СВЕДЕНИЯ О СИСТЕМЕ MATLAB

§2.1. Общие сведения о MATLAB

История создания и развития системы MATLAB (полностью MATrix LABoratory - Матричная Лаборатория) насчитывает более сорока пяти десятков лет. Классический MATLAB был создан Кливом Моулером (университет Нью-Мехико) в 1977 году и являлся интерактивной матричной лабораторией, позволяющей вызывать подпрограммы из пакетов LINPACK и EISPACK. Далее появлялись новые некоммерческие версии MATLAB. В 1984 году К. Моулер, С. Бангерт, Дж. Литтл образовали фирму MathWorks и начиная с этого момента стали выходить коммерческие версии этой системы. Последняя версия MATLAB R2023b вышла в 2023 году. Сейчас MATLAB является мощным математическим пакетом со своим языком программирования, гибкими графическими возможностями, средствами сопряжения с другими языками и несколькими десятками пакетов приложений. В MATLAB используется математический сопроцессор и допускается обращения к программам, которые написаны на языках Fortran, C и C++. Также интерфейс последних версий MATLAB совместим с Python.

Области применения MATLAB весьма обширны. Наиболее известные из них: математические вычисления; проектирование алгоритмов; имитационное моделирование и вычислительный эксперимент; анализ данных, в том числе, интеллектуальный (Data Mining); исследование и визуализация результатов; инженерная графика; разработка приложений; проектирование графического интерфейса пользователя.

MATLAB является интерактивной системой, в которой **основным объектом** является **массив**. Особенностью MATLAB является то, что для массива не требуется указывать размерность в явном виде. При программировании на скалярных языках это позволяет существенно сокращать время решения многих вычислительных задач, связанных с векторно-матричными формулировками. MATLAB ориентирован на работу с реальными данными и поддерживает вычисления с плавающей точкой.

Одновременно, являясь операционной средой, MATLAB имеет свой язык программирования, который позволяет создавать программы для многократного использования. MATLAB предоставляет возможность пользователю создавать специализированные функции, которые также как и программы оформляются в виде М-файлов. Кроме этого, последние версии MATLAB имеют свободно распространяемый независимый набор библиотек общего пользования MATLAB Runtime, который позволяет запускать скомпилированные приложения MATLAB Apps на компьютерах без установки самой MATLAB.

Одним из концептуальных моментов в MATLAB является объединение программ, выполняющих родственные функции, в пакеты прикладных программ (Application Toolboxes), которые включают коллекцию М-файлов, связанных однотипностью решаемых задач или общей проблематикой.

Пакеты прикладных программ Toolboxes, входящие в каждую версию MATLAB, на самом деле являются результатом труда исследователей, объединенных в проблемно ориентированные группы по всему миру. Так были созданы Application Toolboxes для решения различных задач вычислительной

математики (решения дифференциальных уравнений в частных производных, сплайн-аппроксимации, оптимизации и т.д.) и математической статистики; проектирования робастных систем управления; обработки сигналов; структурной и параметрической идентификации линейных и нелинейных динамических систем; разработки нечетких моделей и нейронных сетей; реализации методов эволюционного программирования. Несколько последних версий MATLAB позволяют разрабатывать нейронные сети глубокого обучения различного типа для решения различных прикладных задач. Поэтому пакеты прикладных программ Application Toolboxes, входящие в состав различных версий MATLAB, нашли широкое применение в научных исследованиях, а также разных областях техники.

MATLAB включает пять основных частей:

1. Язык MATLAB является языком высокого уровня с управлением потоками, включающим структуры данных, основанные на матрицах. Содержит широкий спектр функций работы с матрицами и массивами, интегрированную среду разработки, объектно-ориентированные возможности и интерфейсы к программам, написанным на языках Fortran, C, C++ и Python. Возможности, имеющиеся в языке MATLAB, позволяют создавать, как небольшие по масштабу программы, так и большие проекты и сложные приложения.

2. Среда MATLAB включает набор инструментов и приспособлений, которые использует пользователь MATLAB при программировании. Среда MATLAB содержит средства управления вводом и выводом данных, управления переменными в рабочем пространстве, а также работы (создания, контроля и отладки) с М-файлами и приложениями.

3. Графическая система MATLAB. Это управляемая графика, которая содержит широкий набор команд низкого и высокого уровня. Команды высокого уровня используются для отображения двух- и трехмерных данных, работы с изображениями, анимацией и иллюстрированной графикой. Команды низкого уровня используются для полного редактирования внешнего

вида графики, а также создания GUI (Графический интерфейс пользователя) в проектах и приложениях MATLAB.

4. Библиотека математических функций включает обширный набор алгоритмов для вычисления элементарных функций (сумма, синус, косинус, тангенс, котангенс и т.д.); работы с векторами, матрицами и массивами (обращение матриц, нахождение собственных значений, формирования специальных матриц и т.д.); вычисления различных функций (функции Эйри, Бесселя, интегральная показательная функция, гамма-функция и т.д.); быстрое преобразование Фурье и т.д.

5. Программный интерфейс с внешними языками (Application Program Interface) включает библиотеку, позволяющую работать с программами, написанными на Fortran, C, C++ и Python, для организации их взаимодействия с MATLAB. Библиотека обеспечивает динамическую связь для вызова программ из MATLAB. При этом MATLAB используется как вычислительный инструмент и инструмент для чтения-записи MAT-файлов.

§2.2. Пакеты расширения MATLAB

MATLAB включает в свой состав ряд компонентов, который можно просмотреть при помощи команды `ver`. Кроме этого, по команде `ver` выводятся название, номер версии и дата создания:

```
» ver
MATLAB Version: 9.11.0.1769968 (R2021b)
MATLAB License Number: 968398
Operating System: Майкрософт Windows 10 Домашняя для одного языка
Version 10.0 (Build 19045)
Java Version: Java 1.8.0_202-b08 with Oracle Corporation Java
HotSpot(TM) 64-Bit Server VM mixed mode
-----
MATLAB                Version 9.11      (R2021b)
Simulink               Version 10.4     (R2021b)
5G Toolbox             Version 2.3      (R2021b)
AUTOSAR Blockset       Version 2.5      (R2021b)
Aerospace Blockset     Version 5.1      (R2021b)
Aerospace Toolbox      Version 4.1      (R2021b)
Antenna Toolbox        Version 5.1      (R2021b)
Audio Toolbox          Version 3.1      (R2021b)
Automated Driving Toolbox Version 3.4      (R2021b)
Bioinformatics Toolbox Version 4.15.2    (R2021b)
```

Communications Toolbox	Version 7.6	(R2021b)
Computer Vision Toolbox	Version 10.1	(R2021b)
Control System Toolbox	Version 10.11	(R2021b)
Curve Fitting Toolbox	Version 3.6	(R2021b)
DDS Blockset	Version 1.1	(R2021b)
DSP System Toolbox	Version 9.13	(R2021b)
Data Acquisition Toolbox	Version 4.4	(R2021b)
Database Toolbox	Version 10.2	(R2021b)
Datafeed Toolbox	Version 6.1	(R2021b)
Deep Learning HDL Toolbox	Version 1.2	(R2021b)
Deep Learning Toolbox	Version 14.3	(R2021b)
Econometrics Toolbox	Version 5.7	(R2021b)
Embedded Coder	Version 7.7	(R2021b)
Filter Design HDL Coder	Version 3.1.10	(R2021b)
Financial Instruments Toolbox	Version 3.3	(R2021b)
Financial Toolbox	Version 6.2	(R2021b)
Fixed-Point Designer	Version 7.3	(R2021b)
Fuzzy Logic Toolbox	Version 2.8.2	(R2021b)
GPU Coder	Version 2.2	(R2021b)
Global Optimization Toolbox	Version 4.6	(R2021b)
HDL Coder	Version 3.19	(R2021b)
HDL Verifier	Version 6.4	(R2021b)
Image Acquisition Toolbox	Version 6.5	(R2021b)
Image Processing Toolbox	Version 11.4	(R2021b)
Instrument Control Toolbox	Version 4.5	(R2021b)
LTE Toolbox	Version 3.6	(R2021b)
Lidar Toolbox	Version 2.0	(R2021b)
MATLAB Coder	Version 5.3	(R2021b)
MATLAB Compiler	Version 8.3	(R2021b)
MATLAB Compiler SDK	Version 6.11	(R2021b)
MATLAB Report Generator	Version 5.11	(R2021b)
Mapping Toolbox	Version 5.2	(R2021b)
Mixed-Signal Blockset	Version 2.1	(R2021b)
Model Predictive Control Toolbox	Version 7.2	(R2021b)
Model-Based Calibration Toolbox	Version 5.11	(R2021b)
Motor Control Blockset	Version 1.3	(R2021b)
Navigation Toolbox	Version 2.1	(R2021b)
OPC Toolbox	Version 5.0.3	(R2021b)
Optimization Toolbox	Version 9.2	(R2021b)
Parallel Computing Toolbox	Version 7.5	(R2021b)
Partial Differential Equation Toolbox	Version 3.7	(R2021b)
Phased Array System Toolbox	Version 4.6	(R2021b)
Powertrain Blockset	Version 1.10	(R2021b)
Predictive Maintenance Toolbox	Version 2.4	(R2021b)
RF Blockset	Version 8.2	(R2021b)
RF PCB Toolbox	Version 1.0	(R2021b)
RF Toolbox	Version 4.2	(R2021b)

ROS Toolbox	Version 1.4	(R2021b)
Radar Toolbox	Version 1.1	(R2021b)
Reinforcement Learning Toolbox	Version 2.1	(R2021b)
Risk Management Toolbox	Version 1.10	(R2021b)
Robotics System Toolbox	Version 3.4	(R2021b)
Robust Control Toolbox	Version 6.11	(R2021b)
Satellite Communications Toolbox	Version 1.1	(R2021b)
Sensor Fusion and Tracking Toolbox	Version 2.2	(R2021b)
SerDes Toolbox	Version 2.2	(R2021b)
Signal Integrity Toolbox	Version 1.0	(R2021b)
Signal Processing Toolbox	Version 8.7	(R2021b)
SimBiology	Version 6.2	(R2021b)
SimEvents	Version 5.11	(R2021b)
Simscape	Version 5.2	(R2021b)
Simscape Driveline	Version 3.4	(R2021b)
Simscape Electrical	Version 7.6	(R2021b)
Simscape Fluids	Version 3.3	(R2021b)
Simscape Multibody	Version 7.4	(R2021b)
Simulink 3D Animation	Version 9.3	(R2021b)
Simulink Check	Version 5.2	(R2021b)
Simulink Code Inspector	Version 4.0	(R2021b)
Simulink Coder	Version 9.6	(R2021b)
Simulink Compiler	Version 1.3	(R2021b)
Simulink Control Design	Version 6.0	(R2021b)
Simulink Coverage	Version 5.3	(R2021b)
Simulink Design Optimization	Version 3.10	(R2021b)
Simulink Design Verifier	Version 4.6	(R2021b)
Simulink Desktop Real-Time	Version 5.13	(R2021b)
Simulink PLC Coder	Version 3.5	(R2021b)
Simulink Real-Time	Version 7.2	(R2021b)
Simulink Report Generator	Version 5.11	(R2021b)
Simulink Requirements	Version 1.8	(R2021b)
Simulink Test	Version 3.5	(R2021b)
SoC Blockset	Version 1.5	(R2021b)
Spreadsheet Link	Version 3.4.6	(R2021b)
Stateflow	Version 10.5	(R2021b)
Statistics and Machine Learning Toolbox	Version 12.2	(R2021b)
Symbolic Math Toolbox	Version 9.0	(R2021b)
System Composer	Version 2.1	(R2021b)
System Identification Toolbox	Version 9.15	(R2021b)
Text Analytics Toolbox	Version 1.8	(R2021b)
UAV Toolbox	Version 1.2	(R2021b)
Vehicle Dynamics Blockset	Version 1.7	(R2021b)
Vehicle Network Toolbox	Version 5.1	(R2021b)
Vision HDL Toolbox	Version 2.4	(R2021b)
WLAN Toolbox	Version 3.3	(R2021b)
Wavelet Toolbox	Version 6.0	(R2021b)
Wireless HDL Toolbox	Version 2.3	(R2021b)

Множество пакетов расширения, входящих в MATLAB, позволяют усилить функциональные возможности системы, повысить скорость, эффективность и точность вычислений.

Перечень основных пакетов расширения MATLAB 2021b систематизирован в таблице 2.1.

Таблице 2.1 Основные пакеты расширения MATLAB 2021b

№	Наименование	Функциональное назначение
Пакеты для моделирования систем, работы с нечеткой логикой, нейронными сетями и символьной математикой		
1.	SIMULINK	<p>SIMULINK является пакетом расширения, предназначенным для имитационного моделирования систем, составленных из графических блоков с заданными параметрами.</p> <p>Графические блоки и их модели, входящие в модель системы, находятся в разделах библиотеки SIMULINK, управление которой осуществляется через браузер Library Browser.</p> <p>При помощи мыши блоки могут переноситься в основное окно модели и соединяться друг с другом необходимыми связями. В состав моделей могут включаться различные блоки: источники сигналов различного вида, осциллографы, функциональные преобразователи и т.д.</p>
2.	Real Time Windows Target и Workshop	<p>Real Time Windows Target и Workshop являются пакетами расширения, подключающимися к SIMULINK для организации имитационного моделирования в реальном масштабе времени.</p> <p>При наличии дополнительных аппаратных средств (плат расширения компьютера) они представляют собой мощное средство управления объектами и системами, существующими в реальности.</p>
3.	Report Generator для MATLAB и Simulink	<p>Report Generator генерирует отчеты о работе системы MATLAB и пакета расширения SIMULINK. Генератор отчетов очень полезен при моделировании сложных систем, либо отладке сложных вычислительных алгоритмов.</p> <p>Запуск генератора отчетов производится командой Report. Отчеты могут иметь вид программ, для которых предусмотрена возможность редактирования.</p>
4.	Neural Networks Toolbox	<p>Neural Networks Toolbox является пакетом прикладных программ, в котором содержатся средства</p>

		<p>для построения искусственных нейронных сетей. Пакет позволяет осуществлять проектирование, обучение множества известных нейронных сетей, начиная от моделей мультиперсептрона и кончая одними из самых современных ассоциативных и самоорганизующихся сетей.</p> <p>Пакет может быть использован для исследования и применения нейронных сетей к таким задачам, как обработка сигналов, нелинейное управление, прогнозирование, распознавание образов, финансовое моделирование.</p>
5.	Fuzzy Logic Toolbox	<p>Fuzzy Logic Toolbox является пакетом прикладных программ, относящихся к теории нечетких множеств. В пакете обеспечивается возможность построения и тонкой настройки нечетких моделей Мамдани и Сугено.</p> <p>Запуск пакета производится командой <code>fuzzy</code>. Графические средства пакета позволяют интерактивно отслеживать результаты выполнения промежуточных этапов построения и настройки системы нечеткого вывода.</p>
6.	Deep Network Designer	<p>Deep Network Designer является приложением, предназначенным для графического проектирования, анализа и обучения нейронных сетей. Пакет служит основой при разработке и реализации нейронных сетей глубокого обучения. В состав пакета входят алгоритмы, предварительно обученные модели и приложения. Пакет позволяет создавать сверточные нейронные сети (ConvNets, CNNs) и сети долгой краткосрочной памяти (LSTM), которые используются для работы с изображениями, временными рядами и текстами. Также имеется возможность создания генеративных состязательных сетей (GANs), сиамских сетей, пользовательских циклов для обучения и совместно использованных весов.</p>
7.	Symbolic Math Toolbox	<p>Symbolic Math Toolbox - пакет прикладных программ, позволяющий MATLAB решать задачи в символьном виде, реализуя точную арифметику произвольной разрядности.</p> <p>Пакет базируется на применении ядра символьной математики системы компьютерной алгебры Maple V R4 и обеспечивает выполнение ряда операций. В их число входят: символьное дифференцирование и интегрирование, вычисление сумм и произведений, разложение в ряды Тейлора и Маклорена, операции</p>

		над полиномами (степенными многочленами), вычисление корней полиномов, решение в аналитическом виде нелинейных уравнений, всевозможные символьные преобразования, подстановки и целый ряд других операций.
Пакеты математических вычислений		
8.	NAG Foundation Toolbox	NAG Foundation Toolbox является одной из самых мощных библиотек математических функций. Пакет содержит множество новых функций.
9.	Spline Toolbox	Spline Toolbox является пакетом прикладных программ, предназначенным для работы с различными сплайнами. Пакет поддерживает одномерную и многомерную сплайн-интерполяцию, экстраполяцию и аппроксимацию. В пакете обеспечивается представление и отображение сложных данных и поддержка графики.
10.	Statistics Toolbox	Statistics Toolbox является пакетом прикладных программ, предназначенным для реализации статистических вычислений и статистической обработки данных. В пакет входит набор средств для генерации случайных чисел, векторов, матриц и массивов с различными законами распределения, а также множество статистических функций.
11.	Optimization Toolbox	Optimization Toolbox является пакетом прикладных программ для решения оптимизационных задач и систем нелинейных уравнений. Поддерживает различные методы оптимизации функций от множества переменных.
12.	Partial Differential Equations Toolbox	Partial Differential Equations Toolbox является пакетом прикладных программ, предназначенным для решения систем дифференциальных уравнений в частных производных. Пакет использует метод конечных элементов, а его команды и графический интерфейс могут быть использованы для математического моделирования систем с распределенными параметрами в инженерных и научных задачах, включая задачи диффузии, тепло- и массопереноса, сопротивления материалов и т.д.
Пакеты анализа и синтеза систем управления		
13.	Control System Toolbox	Control System - пакет, предназначенный для моделирования, анализа и проектирования непрерывных и дискретных систем автоматического управления.

		<p>В пакете реализуются две группы методов: передаточных функций и пространства состояний. Пакет предоставляет возможность быстрого вычисления и отображения на экране частотных и временных откликов, диаграмм расположения нулей и полюсов. В пакете реализованы:</p> <ul style="list-style-type: none"> • полный набор средств для анализа систем МИМО (множество входов - множество выходов); • временные характеристики: передаточная и переходная функции, реакция на произвольное воздействие; • частотные характеристики: диаграммы Боде, Николса, Найквиста и др.; • разработка обратных связей; • проектирование LQR/LQE-регуляторов; • характеристики моделей: управляемость, наблюдаемость, понижение порядка моделей; • поддержка систем с запаздыванием. <p>Пакет имеет средства выбора параметров обратной связи. Пакет использует, как традиционные, так и современные методы решения задач. Среди традиционных методов можно выделить: анализ особых точек, определение коэффициента усиления и затухания. Современные методы включают: линейно-квадратичное регулирование и др. В пакет входит большое количество алгоритмов, предназначенных для проектирования систем управления. Кроме того, пакет обладает настраиваемым окружением и позволяет создавать свои собственные т-файлы.</p>
14.	Nonlinear Control Design Toolbox	<p>Пакет Nonlinear Control Design Blockset является инструментом, реализующим метод динамической оптимизации при проектировании систем управления.</p> <p>Пакет используется совместно с SIMULINK, производит автоматическую настройку системных параметров, основываясь на ограничениях по временным характеристикам, определенным пользователем. В пакете используется перенос объектов мышью для изменения временных ограничений прямо на графиках. Это позволяет легко настраивать переменные и указывать неопределенные параметры.</p> <p>Пакет обеспечивает интерактивную оптимизацию, реализацию моделирования методом Монте-Карло, поддерживает проектирование SISO и МИМО систем управления, позволяет моделировать подавление помех, слежение и другие типы откликов,</p>


		поддерживает проблемы повторяющегося параметра и задачи управления системами с запаздыванием, позволяет осуществлять выбор между удовлетворенными и недостижимыми ограничениями.
15.	Robust Control Toolbox	<p>Пакет Robust Control является инструментом для проектирования и анализа многопараметрических устойчивых систем управления.</p> <p>Эти системы могут содержать ошибки моделирования, иметь не полностью известную динамику или изменяющиеся по ходу моделирования параметры. Имеющиеся в пакете алгоритмы позволяют выполнять сложные вычисления, учитывающие изменения множества параметров.</p>
16.	Model Predictive Control Toolbox	<p>Пакет Model Predictive Control включает необходимый набор средств для осуществления стратегии упреждающего (предиктивного) управления. Стратегия предиктивного управления используется для управления сложными многоканальными процессами, когда имеются ограничения на переменные состояния и управление. Предикативное управление используется для управления непрерывными процессами в различных отраслях промышленности.</p> <p>Пакет включает более пятидесяти специализированных функций для решения задач, возникающих при проектировании и управлении динамическими системами с использованием предикативного управления. В нем предусмотрена поддержка множества типов систем: импульсных, непрерывных и дискретных по времени, пространство состояний. Используются различные виды возмущений.</p>
17.	(Мю)-Analysis and Synthesis	<p>Пакет p-Analysis and Synthesis включает набор функций для проектирования устойчивых систем управления. В пакете используется оптимизация в равномерной норме и сингулярный параметр.</p> <p>Пакет оснащен графическим интерфейсом, позволяющим упростить выполнение операций при работе с блоками в процессе проектирования оптимальных регуляторов.</p>
18.	Stateflow	<p>Пакет Stateflow является инструментом для моделирования событийно-управляемых систем. Основу пакета составляет теория конечных автоматов.</p> <p>Пакет используется вместе с пакетом моделирования динамических систем SIMULINK. В любую модель, созданную в SIMULINK, можно вставить Stateflow- диаграмму (SF-диаграмму), которая</p>

		<p>является анимационной и отражает поведение компонентов моделируемой системы.</p> <p>По выделяющимся цветом блокам и связям в SF-диаграмме можно проследить все стадии работы моделируемой системы и сделать ее работу зависимой от конкретных событий.</p>
19.	Quantitative Feedback Theory Toolbox	<p>Пакет Quantitative Feedback Theory Toolbox включает функции, предназначенные для создания робастных систем с обратной связью. QFT расшифровывается как количественная теория обратных связей. В основе лежит метод, который использует частотное представление моделей для удовлетворения различных требований к качеству при наличии неопределенных характеристик объекта. Метод базируется на наблюдении того, что обратная связь необходима в тех случаях, когда некоторые характеристики объекта неопределенны и/или на его вход подаются неизвестные возмущения.</p>
20.	LMI Control Toolbox	<p>Пакет LMI (Linear Matrix Inequality) Control является интегрированной средой для решения задач линейного программирования.</p> <p>Первоначально пакет LMI Control Toolbox был предназначен для проектирования систем управления. В настоящее время пакет позволяет решать любые задачи линейного программирования и может быть использован практически в любой сфере деятельности.</p> <p>Пакет имеет следующие основные возможности:</p> <ul style="list-style-type: none"> • исследование задач линейного программирования; • графический редактор задач линейного программирования; • задание ограничений в символьном виде; • многокритериальное проектирование регуляторов; • проверка устойчивости: квадратичная устойчивость линейных систем, устойчивость по Ляпунову, проверка критерия Попова для нелинейных систем. <p>Пакет LMI Control имеет два вида графического интерфейса пользователя:</p> <ul style="list-style-type: none"> • редактор задачи линейного программирования (LMI Editor), который позволяет задавать ограничения в символьном виде; • интерфейс Magshape, обеспечивающий пользователя удобными средствами работы с пакетом.

Пакеты идентификации систем		
21.	System Identification Toolbox	Пакет System Identification является инструментарием для создания математических моделей динамических систем, основанного на известных значениях входных и выходных параметров.
22.	Domain System Identification Toolbox	<p>Пакет Frequency Domain System Identification является инструментарием, предназначенным для идентификации линейных динамических систем по их временному или частотному отклику.</p> <p>Частотные методы применяются при идентификации непрерывных систем и является дополнением к более традиционной дискретной методике.</p> <p>Пакет может быть применен к задачам моделирования динамических процессов, происходящих в электротехнических, механических, гидродинамических системах и т.д.</p>
Дополнительные пакеты расширения MATLAB		
23.	Communications Toolbox	<p>Пакет Communications Toolbox предназначен для построения и моделирования разнообразных телекоммуникационных устройств.</p> <p>Пакет содержит большой набор моделей различных устройств связи и ряд наглядных примеров моделирования телекоммуникационных средств.</p>
24.	Digital Signal Processing (DSP) Blockset	<p>Пакет Digital Signal Processing Blockset предназначен для проектирования устройств, которые используют процессоры цифровой обработки сигналов. Прежде всего к ним относятся высокоэффективные цифровые фильтры с заданной или адаптируемой к параметрам сигналов амплитуда-частотной характеристикой.</p> <p>Пакет позволяет осуществлять моделирование и проектирование цифровых устройств, результаты которых могут быть использованы для построения высокоэффективных цифровых фильтров на базе сигнальных микропроцессоров.</p>
25.	Fixed-Point Blockset	Пакет Fixed-Point Blockset предназначен для моделирования цифровых систем управления и цифровых фильтров совместно с пакетом SIMULINK.
Пакеты для обработки сигналов и изображений		
26.	Signal Processing Toolbox	<p>Пакет Signal Processing Toolbox является мощным средством для анализа, моделирования и проектирования устройств обработки различных сигналов.</p> <p>Пакет обеспечивает фильтрацию и множество преобразований сигналов. В состав пакета входят модули, предназначенные для разработки линейных систем и анализа временных рядов. Пакет может быть полезен в различных областях: обработка мультимедийной информации, телекоммуникации, управление в режиме реального времени, экономика, финансы, медицина и т.д.</p>

27.	Higher-Order Spectral Analysis Toolbox	<p>Пакет Higher-Order Spectral Analysis включает алгоритмы для анализа сигналов, использующие моменты высшего порядка.</p> <p>Пакет содержит наиболее передовые методы анализа и обработки сигналов, что обеспечивает широкие возможности для анализа негауссовых сигналов.</p>
28.	Image Processing Toolbox	<p>Пакет Image Processing предоставляет различные средства для цифровой обработки и анализа изображений, что позволяет использовать его в научных и инженерных задачах, искусстве и т.д.</p>
29.	Wavelet Toolbox	<p>Пакет Wavelet содержит полный набор программ для исследования многомерных нестационарных процессов с помощью коротких волновых пакетов (вейвлетов). Методы пакета Wavelet призваны расширить возможности пользователя в тех областях, где обычно применяется Фурье-разложение.</p> <p>Пакет может быть использован в различных областях: обработка речи и аудиосигналов, телекоммуникации, геофизика, финансы, медицина и т.д.</p>
Прочие пакеты прикладных программ		
30.	Financial Toolbox	<p>Пакет Financial Toolbox содержит прикладные программы для финансово-экономических расчетов.</p> <p>Пакет поддерживает множество функций по моделированию сложных процентов, расчету банковских операций по вкладам, вычислению прибыли и многие другие.</p>
31.	Mapping Toolbox	<p>Пакет Mapping Toolbox служит в качестве графического и командного интерфейса для обработки географических данных, визуализации карт и обеспечения доступа к внешним источникам географических данных. Пакет обеспечивает работу с большим числом широко известных атласов. В совокупности средства пакета создают широкие возможности для эффективной работы с научными географическими данными.</p>
32.	Power System Blockset	<p>Пакет Power System Blockset предназначен для моделирования мощных энергетических систем, включая линий передачи, силовых ключей, регуляторов напряжения и тока, устройств управления электродвигателями различного типа, нагревательных систем и т.д.</p> <p>Пакет обеспечивает моделирование отдельных устройств различной степени сложности и целых электрических систем. Результаты, полученные в ходе моделирования, могут отображаться разнообразными виртуальными измерительными приборами (графопостроители, осциллографы и т.д.).</p>

33.	Data Acquisition Toolbox и Instrument Control Toolbox	<p>Пакет Data Acquisition Toolbox относится к области сбора данных, который осуществляется через блоки и приборы, широко используемые в исследовательских целях для получения данных. Они имеют поддержку необходимой вычислительной базой.</p> <p>Относительно новый блок Instrument Control Toolbox обеспечивает возможность подключения блоков и приборов с последовательным интерфейсом, шиной «Канал общего пользования» и VXI.</p>
34.	Database Toolbox	<p>Пакет Database Toolbox позволяет обеспечить обмен информацией с большим числом систем управления базами данных через драйверы ODBC или JDBC:</p> <ul style="list-style-type: none"> • Oracle 7.3.3; • Access 95 или 97 Microsoft; • Microsoft SQL Server 6.5 или 7.0; • Sybase Adaptive Server 11; • Sybase SQL Server Anywhere 5.0; • IBM DB2 Universal 5.0; • Informix 7.2.2; • Computer Associates Ingres (все версии).
35.	Virtual Reality Toolbox	Пакет Virtual Reality Toolbox предназначен для реализации трехмерной анимации и мультипликации, включая, модели Simulink.
36.	Excel Link	Excel Link является программным макросом, позволяющим интегрировать Microsoft Excel и MATLAB в ОС Windows. Excel Link обеспечивает обмен и синхронизацию данных между этими двумя средами
37.	MATLAB Compiler	MATLAB Compiler является компилятором для программ, написанных на языке программирования системы MATLAB. Транслирует коды этих программ в программы на языке C++. Применение компилятора обеспечивает возможность создания исполняемых кодов, что значительно уменьшает время выполнения программ с большим числом циклических операций.

Вызов MATLAB. Для запуска системы MATLAB в Windows можно использовать двойной щелчок по соответствующей иконке  на рабочем столе. При запуске MATLAB автоматически выполняет М-файлы **matlabrc.m** и **startup.m**. Файл **matlabrc.m**, находящийся в директории **local**, является зарезервированным для компании The MathWorks. В многопользовательских системах он может также использоваться системным менеджером. Файл **startup.m** позволяет задать пользовательские стартовые опции в соответствии с предпочтениями пользователя.

Выход из MATLAB. Для выхода из MATLAB можно выбрать опцию **Exit** (Выход из MATLAB) в меню **File** на рабочем столе MATLAB или выполнить команду **quit** (Выход) в окне командной строки Command Window.

§2.3. Интерфейс системы MATLAB

При запуске MATLAB открывается рабочий стол (desktop) - графический интерфейс пользователя, предоставляющий удобное окружение для работы с системой. На рабочем столе MATLAB располагаются различные окна и панели инструментов, которые предлагают доступ к различным функциям и инструментам системы.

Интерфейс рабочего стола может содержать следующие элементы:

1. В верхней части окна располагается панель инструментов и меню, которое выделено серым цветом заливки и содержит вкладки **HOME**, **PLOTS**, **APPS**. Различные панели инструментов и меню предоставляют доступ к функциям и командам MATLAB, а также возможность32 настройки и управления средой. Начинать работу следует на вкладке **HOME**.
2. Адресную строку, которая указывает рабочий каталог и позволяет выбрать рабочую директорию.
3. **Command Window** (Окно команд). Здесь можно вводить команды MATLAB и видеть результаты их выполнения. Оно также отображает сообщения об ошибках и предупреждения. Двойная стрелка (>>) в окне указывает на начало командной строки.
4. **Current Folder** (Текущая папка). Панель, позволяющая просматривать и управлять файлами и папками в текущей рабочей папке MATLAB.
5. **Workspace** (Рабочее пространство). Здесь отображаются все глобальные переменные, которые определены в текущем сеансе MATLAB выполнения программы, и их значения. Вы можете просматривать и изменять значения переменных в рабочем пространстве.
6. **Editor** (Редактор). Это окно, в котором можно создавать, редактировать и запускать М-файлы MATLAB, которые содержат код и скрипты.

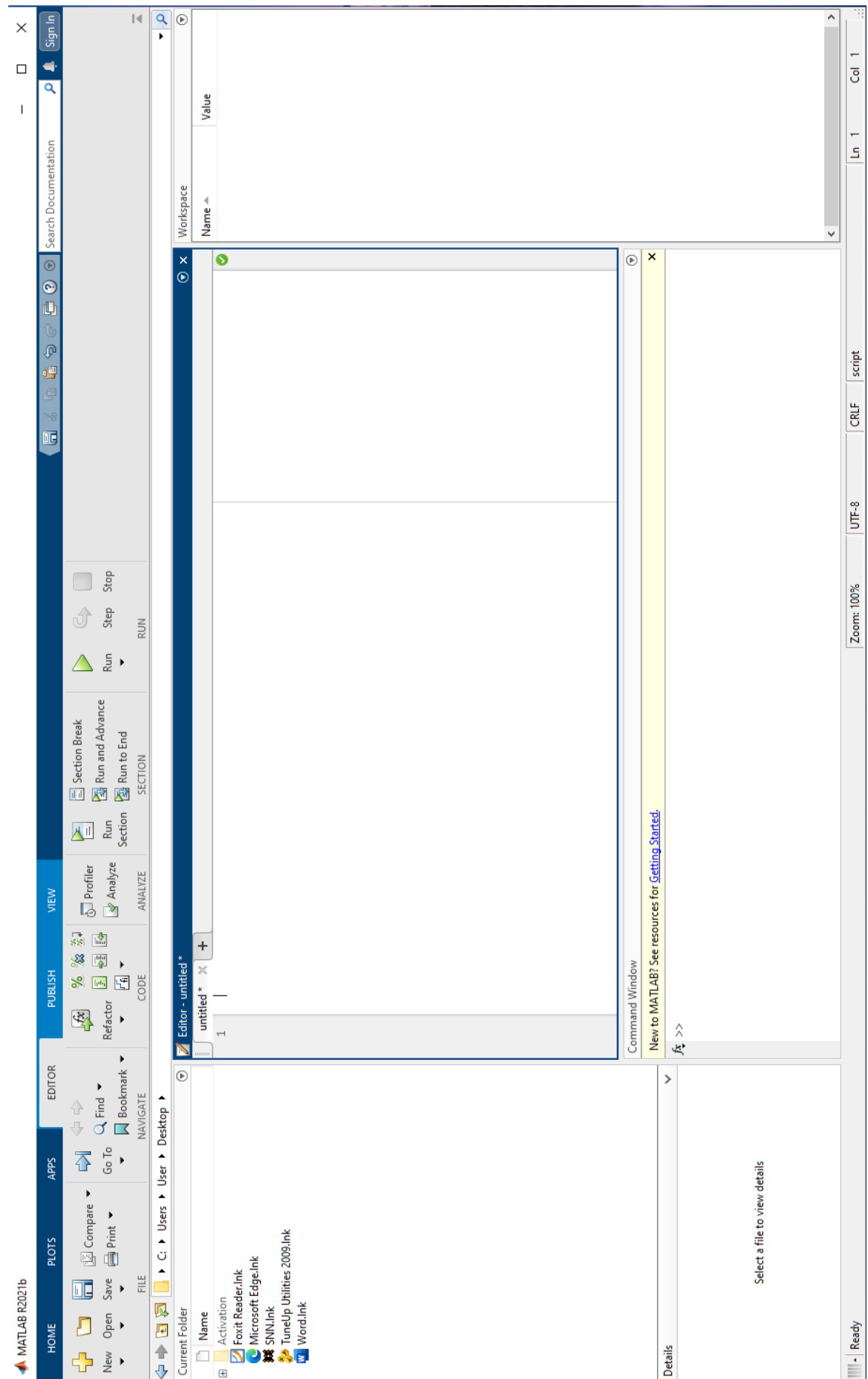


Рисунок 2.1 - Интерфейс системы МАТЛАВ

7. **Figure Window** (Окно графика). Здесь отображаются графики, диаграммы и результаты визуализации данных.
8. **Command History** (Окно истории). Здесь повторяются все команды, выполненные в командной строке.

Настройка набора окон и их расположения может быть осуществлена с использованием кнопки меню **Layout** и путем перемещения окон и их границы мышкой.

Создание новой программы (m- файла) начинается нажатием кнопки **New**, расположенной в верхней левой части меню. При этом происходит открытие окна редактора **Editor**. Окно редактора **Editor** можно встроить в блок окон нажатием на кнопку **Dock** и перемещением его мышкой в необходимое место. Часто такое место находится над **Command Window**.

Другие окна могут появляться в процессе работы программы, например, графики (**Figure**) или интерфейс приложений (например, при вызове **LTI viewer**).

Отдельно следует отметить, что библиотека **SIMULINK** вызывается нажатием кнопки меню **Simulink Library**.

Command Windows (CW)

Все команды в командном окне и их последовательности при интерактивной работе помещаются в строку ввода, которая начинается символом **>>**. Исполнение команды начинается после нажатия клавиши **Enter**. Отделение команд друг от друга осуществляется запятой или точкой запятой. При использовании в качестве разделителя точки с запятой результат выполнения команды не отображается.

На рисунке 2.2 приведен пример выполнения команды *why* в командном окне.

Команды после выполнения помещаются в стек. При помощи стрелок $\uparrow\downarrow$ они могут быть извлечены в строку ввода путем перебора исполненных команд. Такая необходимость возникает при редактировании или повторном исполнении команд. Не доступной для редактирования является строка

вывода. Причина ошибок в MATLAB часто кроется в повторном использовании одних и тех же переменных. Это связано с тем, что все переменные среды в MATLAB являются глобальными. Поэтому желательно каждый раз перед повторным запуском программы очищать содержимое окна **Workspace** командой **Clear Workspace**.

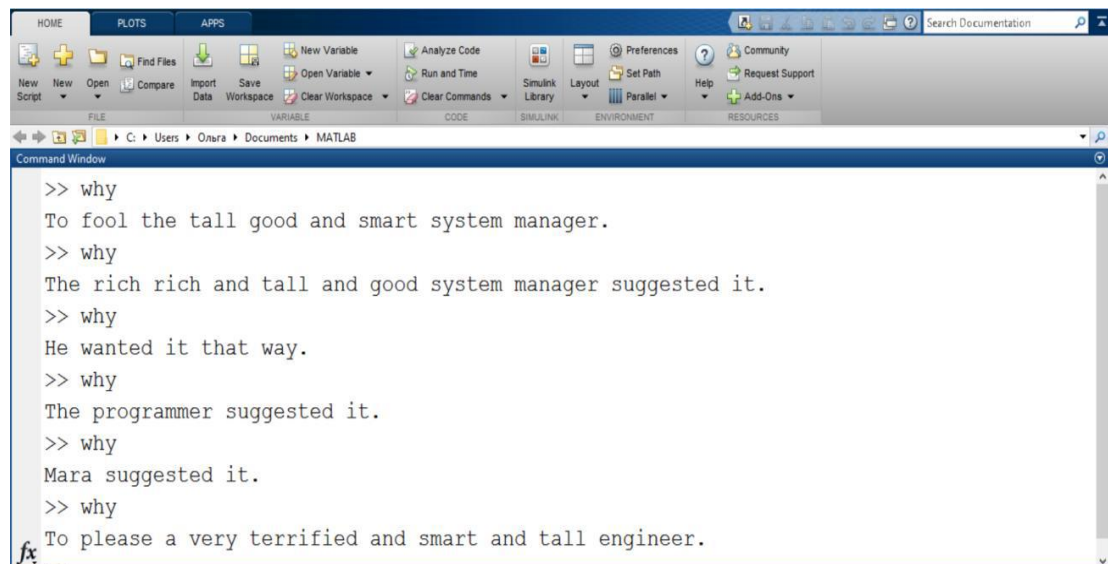


Рисунок 2.2- Результаты исполнения команды *why* в командном окне

Таким образом, окно **Command Windows** одновременно является средой для вычислений и графическим объектом, имеющим тип структуры. Двойственность **Command Windows** сохраняется и в управлении его свойствами.

Workspace

Workspace - это рабочее пространство в MATLAB, которое представляет собой окно, содержащее информацию обо всех переменных, их типах и значениях. В **Workspace** отображаются все переменные, которые были определены или загружены в текущей сессии MATLAB.

Чтобы открыть **Workspace**, можно воспользоваться соответствующим пунктом меню или выполнить команду `workspace` в командном окне.

В **Workspace** каждая переменная представлена пиктограммой, которую можно нажать для активации редактора переменных (**Variable Editor**, VE). Редактор переменных позволяет изменять значения переменных в интерактивном режиме, а также добавлять, удалять или редактировать переменные.

Прием работы с редактором переменных отображен на рисунке 2.3



Рисунок 2.3- Пример работы редактора интерфейса Workspace

Help

Справочная система MATLAB содержит информацию о доступных модулях - Toolbox (Help→Product Help), а также предоставляет алфавитный (Index) и содержательный (Contents) поиск для исследования имеющегося программного функционала. Кроме того, в справочной системе присутствуют демонстрационные программы по различным тематикам (Demos).

Структурирование MATLAB в виде отдельных модулей обусловлено политикой компании MathWorks. Каждый модуль представляет собой изолированную среду для работы в определенной предметной области. Они все ориентированы на использование преимуществ MATLAB, таких как высокая точность, векторно-матричная природа, простой синтаксис и нестрогая типизация.

В справочной системе MATLAB можно использовать контекстный поиск (Help→Product Help→Contents), чтобы найти статью о «Program Control Statements», где описаны все элементы программирования, используемые в MATLAB. Алфавитный поиск (Help→Product Help→Index) позволяет найти информацию о контроле возможных типов данных, введя запрос is*.

На рисунке 2.4 показано окно справочной системы MATLAB. Как и во многих других приложениях, в справочной системе доступны примеры, которые можно скопировать и выполнить в рабочей среде MATLAB.

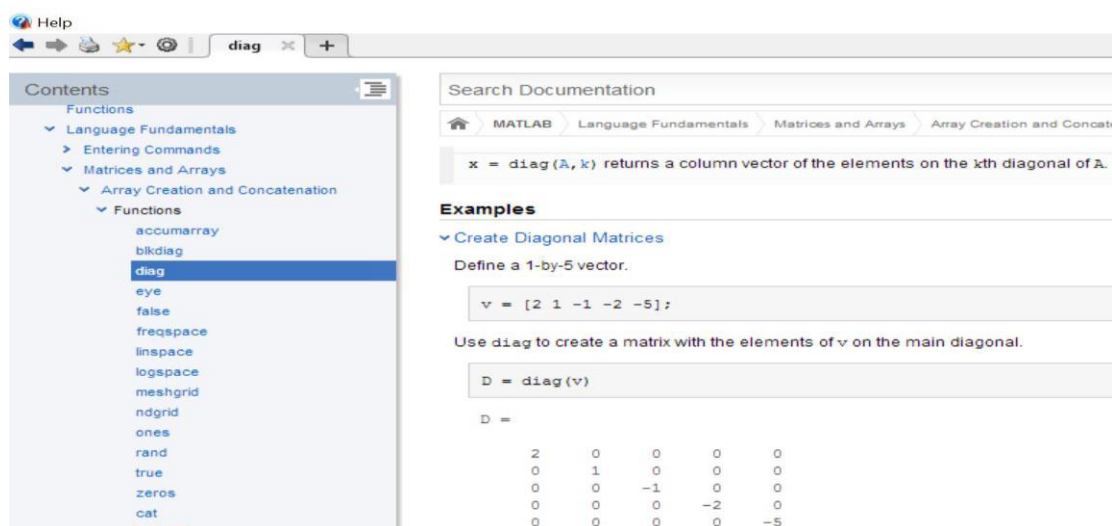


Рисунок 2.4- Окно справочной системы MATLAB

Контрольные вопросы

1. Что такое MATLAB. Назначение.
2. Основной элемент данных MATLAB.
3. Понятие Toolboxes.
4. Пять основных частей системы MATLAB.
5. Что называется пакетом расширения MATLAB.
6. Способ вывода списка пакетов расширения.
7. Основное назначение пакета расширения SIMULINK.
8. Пакеты математических вычислений. Их особенности.
9. Пакеты анализа и синтеза систем управления. Состав.
10. Что такое MIMO, MISO, SISO - системы управления.
11. Пакет Control System Toolbox: назначение и функции.
12. Назовите другие пакеты расширения.

ГЛАВА 3. ОСНОВНЫЕ ВЫРАЖЕНИЯ И ОПЕРАЦИИ

§3.1. Математические выражения

Выражения в MATLAB состоят из переменных, чисел, операторов и функций.

Переменная в MATLAB может быть массивом (матрицей), вектором или скаляром. Одной из преимуществ MATLAB является то, что он не требует предварительного объявления типа переменной или указания размерности

массива. Имена переменных, констант и функций могут состоять из символов латинского алфавита (кроме специальных символов), цифр и символа подчеркивания '_', начинаться должны с буквы. При идентификации переменной используются первые 31 символ имени и MATLAB различает регистр символов в именах переменных.

Рекомендуется использовать осмысленные имена переменных при программировании, чтобы облегчить понимание кода.

В MATLAB также доступны системные константы, включающие:

pi = 3.14159265358979, представляющую значение числа π ;

i, представляющую мнимую единицу. Также для обозначения мнимой единицы можно использовать символ **j**;

Inf, представляющую бесконечность, которая получается в результате деления на ноль;

NaN представляющую неопределенное значение (Not a Number), возникающее, например, в результате деления нуля на ноль 0/0, или вычитания бесконечностей inf-inf;

eps = 2^{-52} или 2.2204e-016, которая характеризует точность вычислений с плавающей точкой в MATLAB.

Основные типы данных в MATLAB. Типы данных, доступные в MATLAB, представлены в табл. 3.1.

Таблица 3.1

Основные типы данных в MATLAB

double	вещественный, 64 бит
single	вещественный, 32 бит
int8	знаковый целочисленный, 8 бит
int16	знаковый целочисленный, 16 бит
int32	знаковый целочисленный, 32 бит
int64	знаковый целочисленный, 64 бит
uint8	беззнаковый целочисленный, 8 бит
uint16	беззнаковый целочисленный, 16 бит
uint32	беззнаковый целочисленный, 32 бит
uint64	беззнаковый целочисленный, 64 бит

По умолчанию MATLAB использует тип данных double, который обеспечивает наибольшую точность представления вещественных чисел и

является универсальным типом. Однако, если требуется экономия памяти, можно явно указать желаемый тип данных следующим образом:

```
a = uint32(989.54); % Задание 32-битного беззнакового  
                    % целочисленного типа  
b = int32(-5.54); % Задание 32-битного знакового  
                  % целочисленного типа
```

§3.2. Арифметические операции с простыми переменными

Базовыми арифметическими операциями над двумя переменными в MATLAB считаются операции, которые записываются следующим образом:

```
c = a+b; %сложение переменных a и b;  
c = a-b; %вычитание переменной b из переменной a;  
c = a*b; %умножение переменной a на переменную b;  
c = a/b; %деление переменной a на переменную b
```

Дополнительно к этим операциям в MATLAB введена операция возведения в степень, которая записывается следующим образом:

```
c = a^3; %возведение переменной a в куб;  
c = a^0.5; %извлечение квадратного корня из переменной a
```

MATLAB также имеет возможность оперировать с комплексными числами в соответствии с арифметикой комплексных чисел. Например, результатом извлечения квадратного корня из -1 будет комплексное число:

```
c = (-1)^0.5; %c = 0.0000 + 1.0000i
```

Для задания комплексных переменных необходимо указать значения их действительной и мнимой частей:

```
a = 5 + 6i; %комплексная переменная a  
b = 3 + 5i; %комплексная переменная b
```

Затем с заданными комплексными переменными также можно выполнять арифметические операции, описанные выше.

MATLAB имеет несколько специальных функций для работы с комплексными числами:

real(x) - извлечение действительной части комплексного числа x;

imag(x) - извлечение мнимой части комплексного числа x;

abs(x) - абсолютное значение комплексного числа x;

conj(x) - определение комплексно-сопряженного числа x;

angle(x) - определение аргумента комплексного числа x.

§3.3. Логические операторы

MATLAB содержит несколько логических операторов:

and (and (a, b))- логическое И ($a \wedge b$, $a \& b$);

or (or (a, b))- логическое ИЛИ ($a \vee b$, $a | b$);

not (not (a))- логическое НЕ ($\neg a$, $\sim a$);

xor (xor (a, b))- исключающее ИЛИ ($a \oplus b$);

any (any (a))- истинно (true), если хотя бы один элемент вектора a не равен нулю;

all (all (a)) - истинно (true), если все элементы вектора a не равны нулю.

§3.4. Основные математические функции

MATLAB содержит большое число математических функций, обращение к которым производится по их имени. Основные математические функции, наиболее распространенные на практике, приведены в таблице 3.2.

Таблица 3.2 Основные математические функции MATLAB

sqrt(x)	вычисление квадратного корня
exp(x)	возведение в степень числа e
pow2(x)	возведение в степень числа 2
log(x)	вычисление натурального логарифма
log10(x)	вычисление десятичного логарифма
log2(x)	вычисление логарифма по основанию 2
sin(x)	синус угла x, заданного в радианах
cos(x)	косинус угла x, заданного в радианах
tan(x)	тангенс угла x, заданного в радианах
cot(x)	котангенс угла x, заданного в радианах
asin(x)	арксинус
acos(x)	арккосинус
atan(x)	арктангенс
pi	pi
round(x)	округление до ближайшего целого
fix(x)	усечение дробной части числа
floor(x)	округление до меньшего целого
ceil(x)	округление до большего целого
mod(x)	остаток от деления с учётом знака
sign(x)	знак числа
factor(x)	разложение числа на простые множители
isprime(x)	истинно, если число простое

rand	генерация псевдослучайного числа с равномерным законом распределения
randn	генерация псевдослучайного числа с нормальным законом распределения
abs(x)	вычисление модуля числа

Большинство приведенных в табл. 3.2 элементарных функций применимы к вычислениям с комплексными аргументами.

Контрольные вопросы

1. Что входит в математические выражения в MATLAB?
2. Что является переменной в MATLAB?
3. Как задаются имена переменных, констант и функций в MATLAB?
4. Назовите системные переменные MATLAB.
5. Назовите основные типы данных MATLAB.
6. Опишите линейную алгоритмическую конструкцию.
7. Какой тип данных используется по умолчанию в MATLAB?
8. Назовите базовые арифметические операции MATLAB.
9. Назовите дополнительные арифметические операции MATLAB.
10. Какие специальных функций используются при работе с комплексными числами?
11. Назовите логические операторы MATLAB.
12. Назовите основные математические функции MATLAB.

ГЛАВА 4. ОПЕРАЦИИ НАД МАТРИЦАМИ И ВЕКТОРАМИ

§4.1. Вектора и матрицы в MATLAB

Вектор в MATLAB задается квадратными скобками. Вектор с именем *a*, содержащий значения 2, 3, 4, 5, имеет следующий вид:

```
a = [2 3 4 5]; % вектор-строка
```

Доступ к нужному элементу вектора осуществляется при помощи следующей конструкции языка:

```
disp(a(1)); %выводится значение 1-го элемента вектора;
disp(a(2)); %выводится значение 2-го элемента вектора;
disp(a(3)); %выводится значение 3-го элемента вектора;
```

```
disp(a(4)); %выводится значение 4-го элемента вектора
```

Определение размера вектора (общего числа элементов), осуществляется функцией **length()** следующим образом:

```
a=[1 2 3 4 5]
```

```
N = length(a); %(N=5) число элементов вектора a
```

Вектор-столбец можно задать двумя способами:

```
a = [1; 2; 3; 4; 5]; %вектор-столбец;
```

```
a = [1 2 3 4 5]'; %вектор-столбец
```

Доступ к элементам вектора-столбца осуществляется так же, как и для вектора-строки.

Векторы можно включать не только отдельные числа или переменные, но и вектора:

```
a = [1 2 3 4 5 6];
```

```
b = [a 10 12]; %второй вектор b = [1 2 3 4 5 6 10 12]
```

Полученный вектор **b** создан на основе вектора **a** и состоит из восьми элементов.

MATLAB имеет функции заполнения векторов нулями, единицами или случайными значениями:

```
a=zeros(1,50);%вектор-строка, 50 элементов с  
%нулевыми значениями;
```

```
b=zeros(50,1);%вектор-столбец, 50 элементов с  
%нулевыми значениями;
```

```
c=ones(1,100);%вектор-строка, 100 элементов с  
%единичными значениями;
```

```
d=ones(100,1);%вектор-столбец, 100 элементов с  
%единичными значениями;
```

```
e=rand(100,1);%вектор-столбец, 100 элементов со  
%случайными значениями
```

Матрицы в MATLAB задаются по аналогии с векторами. Разница заключается лишь в том, что указываются обе размерности:

```
E = [2 0 0; 0 5 0; 0 0 6]; % матрица 3x3,
```

или

```
E = [2 0 0
```

```
0 5 0
```

```
0 0 6]; % матрица 3x3
```


Функции **zeros()**, **ones()** и **rand()** можно использовать аналогичным образом:

```
A = zeros(5,5); %матрица 5x5,заполненная нулями;  
B = zeros(5); % матрица 5x5,заполненная нулями;  
C = ones(5); %матрица 5x5, заполненная единицами;  
D = rand(50);% матрица 50x50, заполненная случайными числами
```

Доступ к элементам матрицы осуществляется командой **disp** с указанием номера строки и столбца, в которых находится требуемый элемент:

```
A = [4 5 6;7 8 9;10 11 12]; %матрица 3x3;  
disp(A(3,3) );% элемент, стоящий в третьей строке  
%третьего столбца, т.е. 12;  
disp(A(1,2) ); % элемент, стоящий в  
%первой строке второго столбца, т.е. 5
```

MATLAB включает операции выделения указанной части матрицы:

```
B = A(:,1); %B = [4; 7; 10] - выделение первого столбца;  
C = A(1,:); %C = [4 5 6] - выделение первой строки;  
D = A(1:2,2:3); %D = [5 6; 8 9] - выделение первых  
%двух строк и 2-го и 3-го столбцов матрицы A
```

Единичная матрица может быть задана функцией **eye()**. Аргументы функции **eye(n,m)** задают размерность матрицы с единицами на основной диагонали и нулями в остальных местах.

```
A=eye(5,5) % матрица A  
% 1      0      0      0      0  
% 0      1      0      0      0  
% 0      0      1      0      0  
% 0      0      0      1      0  
% 0      0      0      0      1
```

Размерность матриц или векторов можно определить с помощью функции **size()**. Функция возвращает число строк и столбцов переменной, указанной в качестве аргумента:

```
a = 6; % переменная a;  
A = [2 3 4 5]; % вектор-строка;  
B = [1 2 3 4; 5 6 7 8]; % матрица 2x4;  
size(a) % 1x1;  
size(A) % 1x4;  
size(B) % 2x4;
```

Возвращение квадратной матрицы с элементами A на n -й диагонали или элементами A на главной диагонали может быть осуществлено функциями **diag(A, n)** или **diag(A)**:

```
A=[1;2;3;4;5]
E=diag(A) % E = 1 0 0 0 0
               %0 2 0 0 0
               %0 0 3 0 0
               %0 0 0 4 0
               %0 0 0 0 5

A=[1;2;3;4]
E=diag(A,4) % E =0 0 0 0 1 0 0 0
               %0 0 0 0 0 2 0 0
               %0 0 0 0 0 0 3 0
               %0 0 0 0 0 0 0 4
               %0 0 0 0 0 0 0 0
               %0 0 0 0 0 0 0 0
               %0 0 0 0 0 0 0 0
               %0 0 0 0 0 0 0 0
```

§4.2. Операции над матрицами и векторами

Достаточно просто в MATLAB выполняются операции сложения и умножения матриц и векторов. Умножение вектор-строки на вектор-столбец дает число, а умножение вектор-столбца на вектор-строку дает двумерную матрицу, которые вычисляются по формулам:

$$c = \sum_{i=1}^5 a_i b_i;$$

$$d = \begin{bmatrix} a_1 b_1 & a_2 b_1 & a_3 b_1 & a_4 b_1 & a_5 b_1 \\ a_1 b_2 & a_2 b_2 & a_3 b_2 & a_4 b_2 & a_5 b_2 \\ a_1 b_3 & a_2 b_3 & a_3 b_3 & a_4 b_3 & a_5 b_3 \\ a_1 b_4 & a_2 b_4 & a_3 b_4 & a_4 b_4 & a_5 b_4 \\ a_1 b_5 & a_2 b_5 & a_3 b_5 & a_4 b_5 & a_5 b_5 \end{bmatrix}.$$

Например, пусть даны два вектора a и b :

```
a = [3 2 3 4 5]; % вектор-строка;
b = [3; 2; 1; 2; 1]; % вектор-столбец;
```

Тогда умножение этих двух векторов можно записать так

```
c = a*b; % c=3*3+2*2+3*1+4*2+5*1=29;
```

```
d = b*a; % d – матрица 5x5 элементов;
      %9      6      9      12      15
      %6      4      6      8      10
      %3      2      3      4      5
      %6      4      6      8      10
      %3      2      3      4      5
```

Сложение и вычитание двух векторов дают вектор той же размерности и выполняется по следующим правилам:

```
a = [1 2 3 4 5];
b = [5 4 3 2 6];
c = a+b; %c = [1+5,2+4,3+3,4+2,5+6] = [6,6,6,6,11];
d = b-a; %d = [5-1,4-2,3-3,2-4,6-5] = [4,2,0,-2,1];
```

Операции сложения и вычитания в MATLAB можно выполнять только между несколькими однотипными векторами: несколькими векторами-столбцами или векторами-строками. При этом, векторы должны иметь одинаковую размерность. Иначе MATLAB выдает сообщение об ошибке.

Аналогично выполняются операции умножения и сложения между матрицами:

```
A = [5 2 3; 4 5 6; 7 8 9];
B = ones(3); %B=[1,1,1;1,1,1;1,1,1];
C= A+B; %сложение двух матриц одинакового размера;
      %C=[6,3,4;5,6,7;8,9,10];
D= A+5; %сложение матрицы и числа;
      %D=[10,7,8;9,10,11;12,13,14];
E = A*B; % умножение матрицы A на B;
      %E=[10,10,10;15,15,15;24,24,24];
F = B*A; % умножение матрицы B на A;
      %F=[16,15,18;16,15,18;16,15,18];
G = 5*A; %умножение матрицы на число;
      %G=[25,10,15;20,25,30;35,40,45]
```

MATLAB включает операции вычисления обратной матрицы, а также транспонирования матриц и векторов. Операции транспонирования матриц и векторов соответствует символ ' (апостроф), который ставится после имени вектора или матрицы. Вычисление обратной матрицы производится функцией **inv()** или возведением матрицы в степень -1.

```
a = [5 5 5]; % вектор-строка;
b = a'; %вектор-столбец, образованный транспонированием
```

```

%вектора-строки a;
A = [4 5 6; 7 8 9; 10 11 12]; %матрица 3x3 элемента;
B = a*A; %B = [105 120 135] - вектор-строка;
C = A*b; %C = [75; 120; 165] - вектор-столбец;
D = a*A*a'; %D = 1800 - число, сумма эл-ов матрицы A;
E = A'; %E - транспонированная матрица A;
%4      7      10
%5      8      11
%6      9      12
F = inv(A); %F - обратная матрица A;
%1.0e+15 *
%-0.9382      1.8765      -0.9382
%1.8765      -3.7530      1.8765
%-0.9382      1.8765      -0.9382
G = A^-1; %G - обратная матрица A
%1.0e+15 *
%-0.9382      1.8765      -0.9382
%1.8765      -3.7530      1.8765
%-0.9382      1.8765      -0.9382

```

MATLAB включает операторы поэлементного умножения `.*`, деления слева направо `./`, деления справа налево `.\`, возведения в степень `.^` векторов или матриц.

```

A=[3 4; 5 6];
B=[1 -3; 4 5];
C=A.*B; %C=[ 3 -12; 20 30];
D=A./B; %D=[ 3 -1.3333; 1.25 1.2];
E= A.\B; %E=[ 0.3333 -0.75; 0.8 0.8333];
A = [5 2 3; 4 5 6; 7 8 9]; %матрица 3x3 элемента;
X=A.^2; % X=[ 25 4 9; 16 25 36; 49 64 81]

```

Поиск максимального значения элемента вектора в MATLAB осуществляется стандартной функцией **max()**, которая может определять не только максимальное значение элемента, но и его позицию. При работе с матрицами данная функция определяет максимальные значения, стоящие в столбцах.

```

a = [5 14 12 4];
b = max(a); % b = 14;
[b, i] = max(a); % b = 14, i = 2;
A = [4 14 5; 6 12 2; 3 15 8];

```

```
B = max(A); % B=[ 6 15 8];
[B, I] = max(A); % B=[ 6 15 8], I = [2 3 3]
```

Минимальное значение элемента вектора или матрицы и его индекс определяются функцией **min()**, которая работает аналогично функции **max()**.

```
a = [5 14 12 4];
b = min(a); % b = 4;
[b, i] = min(a); % b = 4, i = 4;
A = [4 14 5; 6 12 2; 3 15 8];
B = min(A); % B=[ 3 12 2];
[B, I] = min(A); % B=[ 3 12 2], I = [3 2 2]
```

Сумма значений элементов вектора или столбцов матрицы в MATLAB определяется функцией **sum()**:

```
a = [5 14 12 4 1];
s = sum(a); %s = 5+14+12+4+1=36;
A = [4 14 5; 6 12 2; 3 15 8];
S1 = sum(A); %S1=[ 13 41 15];
S2 = sum(sum(A)); %S2=69
```

Функция **sort()** осуществляет сортировку значений элементов вектора или матрицы по возрастанию или убыванию:

```
a = [5 14 12 4 1];
b1 = sort(a); % b1=[ 1 4 5 12 14];
b2 = sort(a, 'descend'); % b2=[ 14 12 5 4 1];
b3 = sort(a, 'ascend'); % b3=[ 1 4 5 12 14];
%для матриц
A = [4 14 5; 6 12 2; 3 15 8];
B1 = sort(A); %B1= [3,12,2; 4,14,5; 6,15,8];
B2 = sort(A, 'descend'); %B2= [6,15,8; 4,14,5; 3,12,2]
```

Функция **find()** осуществляет поиск определенного элемента в векторе или матрице. Аргументом функции является условие, в соответствии с которым находятся требуемые элементы:

```
a = [5 14 12 4 1];
b1 = find(a == 12); %b1 = 3 - индекс элемента 12;
b2 = find(a ~= 12); %b2 = [1 2 4 5] - индексы без 12;
b3 = find(a > 4); %b3 = [1 2 3]
```

Символ '=' в примере означает проверку на равенство, а символ '~=' соответствует проверке на неравенство значений элементов вектора a.

Среднее арифметическое значение при работе с векторами и матрицами вычисляется функцией **mean()**:

```
a = [5 14 12 4 1];  
m = mean(a); % m = 7.2;  
A = [4 14 5; 6 12 2; 3 15 8];  
M1 = mean(A); % M1 = [4.3333 13.6667 5.0];  
M2 = mean(mean(A)); % M2 = 7.6667
```

Возвращение произведения элементов вектора, вектор-строки, содержащей произведения элементов каждого столбца матрицы осуществляется функцией **prod()**:

```
A=[5 14 12 4 1];  
B=prod(A); % B = 3360;  
A5=[4 14 5 12; 6 12 2 15; 3 15 8 34];  
B5=prod(A5); % вектор-строка B5 = 72 2520 80 6120;
```

Возвращение произведения с накоплением в MATLAB осуществляется функцией **cumprod()**. Функция **cumprod(A)** возвращает вектор, содержащий произведения с накоплением элементов вектора A, если A - вектор. Функция **cumprod(A)** возвращает матрицу того же размера, что и A, содержащую произведения с накоплением для каждого столбца матрицы A, если A - матрица:

```
A=[1 2 3 4 5 6];  
B = cumprod(A) % B = 1 2 6 24 120 720  
A5=[1 2 3; 4 5 6; 7 8 9];  
B5 = cumprod(A5); %B5 =  
% 1      2      3  
% 4     10     18  
% 28    80    162
```

Вычисление скалярного произведения векторов и матриц осуществляется функцией **dot()**:

```
a=[1 2 3];  
b=[4 5 6];  
e=dot(a,b) % Скалярное произведение e = 32
```

Вычисление векторного произведения векторов осуществляется функцией **cross()**:

```
a=[1 2 3];  
b=[4 5 6];  
e=cross(a,b) % Векторное произведение e = -3 6 -3
```

Вычисление определителя квадратной матрицы осуществляется функцией **det()**:

```
A=[1 2 0;0 4 5; 6 0 8]
det(A) % Определитель матрицы A равен 92
```

Определение ранга матрицы осуществляется функцией **rank()**:

```
A=[1 2 0;0 4 5; 6 0 8]
rank(A) % Ранг матрицы A равен 3
```

Возвращение различных видов норм матрицы осуществляется функцией **norm()** в зависимости от аргумента p (p=1, 2 inf, fro):

```
A=[1 2 0;0 4 5; 6 0 8]
norm(A) %Максимальное сингулярное значение матричного A
%10.9953
norm(A,1) %Максимальная абсолютная сумма столбца матрицы
%13
norm(A,2) %Эквивалентно максимальному сингулярному
%значению матрицы 10.9953
norm(A,inf) %Максимальная абсолютная сумма
%строки матрицы 14
norm(A,'fro') %Норма Фробениуса
%матричного A 12.0830
```

Функция **cond ()** возвращает число обусловленности матрицы, основанное на норме p:

```
A=[1 2 3;10 4 5; 6 5 8]
cond(A) % число обусловленности матрицы 70.5572
cond(A,1) % число обусловленности матрицы 94.0667
cond(A,2) % число обусловленности матрицы 70.5572
cond(A,'inf') % число обусловленности матрицы 107.6667
cond(A,'fro') % число обусловленности матрицы 72.8103
```

Объединение матриц осуществляется функцией **cat ()**:

```
A=[1 2 0;0 4 5; 6 0 8];
B=[1 5 0;9 4 5; 6 9 8];
cat(2,A,B) % Объединенная матрица 1 2 0 1 5 0
%0 4 5 9 4 5
%6 0 8 6 9 8
```

Возвращение вектора собственных значений матрицы осуществляется функцией **eig()**. Вызов функции в формате [X, Z]=eig(A) даст матрицу X,

столбцы которой - собственные векторы матрицы A, и диагональную матрицу Z, содержащую собственные значения матрицы A:

```
A=[1 2 3;1 4 5; 6 5 8];
[X, Z]=eig(A)
% X = -0.2865 -0.4503 0.0170
%-0.4877 -0.6039 -0.8328
%-0.8247 0.6577 0.5533%
%Z = 13.0414 0 0
%0 -0.6993 0
%0 0 0.6579
```

Возвращение решения системы линейных уравнений $A \cdot x = b$ осуществляется функцией `linsolve()`.

Вызов в формате `linsolve(A, b, options)` позволяет задать метод решения уравнения. Если задать функцию в виде `[x, r]=linsolve(A, b)`, то она вернет x - решение системы и r - ранг матрицы A:

```
A=[1.5 3.6 -0.5;0.6 5.6 10;-0.4 5.4 8.2 ];
b=[4.6;7.8;5.6];
[x, r]=linsolve(A, b)
%x = 1.5159 0.6883 0.3036
%r = 0.0589
```

Приведение матрицы к треугольной форме методом исключений Гаусса осуществляется функцией `rref()`:

```
A=[1.5 3.6 -0.5;0.6 5.6 10;-0.4 5.4 8.2;2 4 6 ];
rref(A) %1 0 0
%0 1 0
%0 0 1
%0 0 0
```

Возвращение разложения по Халецкому для положительно определенной симметрической матрицы осуществляется функцией `chol()`:

```
A = [1 -1 0; -1 5 0; 0 0 7]
chol(A) % 1.0000 -1.0000 0
% 0 2.0000 0
% 0 0 2.6458
```

Выполнение LU-разложения, возвращение двух матриц: нижней треугольной L и верхней треугольную U осуществляется функцией `lu()`:


```

A = [10 -7 0;-3 2 6;5 -1 5];
[L,U] = lu(A)
%L =      1.0000      0      0
      %-0.3000     -0.0400     1.0000
      %0.5000      1.0000      0
%U = 10.0000     -7.0000      0
      %0      2.5000     5.0000
      %0      0      6.2000

```

Выполнение QR - разложения, возвращение ортогональной матрицы Q и верхней треугольной R осуществляется функцией **qr()**:

```

A = [10 -7 0;-3 2 6;5 -1 5];
[Q,R] = qr(A)
%Q =     -0.8639      0.4256     -0.2694
      %0.2592     -0.0831     -0.9623
      %-0.4319     -0.9011     -0.0385
%R =    -11.5758      6.9973     -0.6047
      %0      -2.2444     -5.0042
      %0      0      -5.9660

```

Контрольные вопросы

1. Какой вид имеет запись вектора в MATLAB?
2. Какая конструкция используется для доступа к нужному элементу вектора в MATLAB?
3. Какая функция используется для определения размера вектора?
4. Какие функции используются для инициализации векторов нулями, единицами или случайными значениями в MATLAB?
5. Как задается матрица в MATLAB?
6. Как можно определить размерность вектора или матрицы в MATLAB?
7. Опишите операции сложения и умножения матриц и векторов в MATLAB.
8. Опишите операции вычисления обратной матрицы, а также транспонирования матриц и векторов в MATLAB.
9. Опишите операции поэлементного умножения, деления или возведения в степень элементов вектора или матрицы в MATLAB.

10. Как осуществляется поиск максимального значения элемента вектора в MATLAB?
11. Как осуществляется поиск минимального значения элемента вектора в MATLAB?
12. Как осуществляется вычисление суммы значений элементов вектора или столбцов матрицы в MATLAB?
13. Как осуществляется сортировка значений элементов вектора или матрицы по возрастанию или убыванию в MATLAB?
14. Как осуществляется поиск определенного элемента в векторе или матрице в MATLAB?
15. Как осуществляется вычисление среднего арифметического значения при работе с векторами и матрицами в MATLAB?
16. Что выполняет функция `prod()` в MATLAB?
17. Какая функция возвращает произведение с накоплением в MATLAB?
18. Какая функция используется для вычисления скалярного произведения векторов и матриц в MATLAB?
19. Какая функция используется для вычисления векторного произведения векторов в MATLAB?
20. Назовите функцию вычисления определителя квадратной матрицы.
21. Какая функция используется для возвращения различных видов норм матрицы?
22. Что выполняет функция `cond ()` в MATLAB?
23. Какая функция используется для объединения матриц?
24. Назовите функцию возвращения вектора собственных значений матрицы.
25. Какая функция используется для решения системы линейных уравнений в MATLAB?
26. Какая функция используется для приведения матрицы к треугольной форме методом исключений Гаусса в MATLAB?

ГЛАВА 5. ОРГАНИЗАЦИЯ ВЕТВЛЕНИЯ И ЦИКЛОВ

§5.1. Условные операторы

Организация условных переходов в алгоритмах и создание итерационных (рекуррентных) вычислений в MATLAB осуществляется операторами условных переходов **if**, **else**, **elseif**, **switch**, **case**, **otherwise** и циклов **while**, **for**.

Синтаксис ветвления оператора **if** имеет несколько видов.

1) Оператор с единственным выбором **if** <логическое выражение> <операторы> **end**. Группа инструкций <операторы> выполняется в том случае, когда значение параметра <логическое выражение> соответствует значению «истинно» (**true**, 1). Иначе, когда <логическое выражение> соответствует значению «ложь» (**false**, 0), группа инструкций <операторы> программой пропускается.

```
x=10;  
if x>0  
    y=100  
end  
% y=100
```

2) Оператор с двойным выбором **if** <логическое выражение> <операторы1> **else** <операторы2> **end**. Если значение <логическое выражение> «истинно» (**true**, 1) - выполняется группа инструкций <операторы1>, после чего программа продолжает выполняться, начиная с последующей **end** строки. Иначе происходит выполнение группы инструкций <операторы2>, после чего программа переходит на строку, стоящую после **end**.

```
x=10;  
if x>5  
    y=100  
else  
    y=1000  
end  
% y=1000
```

3) Оператор с тройным и более выбором **if** <логическое выражение1> <операторы1> **elseif** <логическое выражение2> <операторы2> **else** <операторы3> **end**. В операторе вычисляется <логическое выражение1>, стоящее

после ключевого слова **if**. Если значение <логическое выражение1> «истинно» (**true**, 1) - выполняется соответствующая группа инструкций < операторы1>, после чего продолжается выполнение программы, начиная со строки, находящейся после **end**. Если < логическое выражение1> «ложно» (**false**, 0), вычисляется < логическое выражение2>, стоящее после ключевого слова **elseif**. Если <логическое выражение2> окажется истинным, будет выполнена группа инструкций < операторы2>, иначе - будет выполнена группа инструкций < операторы3>, стоящие после ключевого слова **else**. Оператор **elseif** может многократно использоваться внутри оператора условия **if**.

```
x = 4;
y = 5;
if x > y
disp('x больше y');
elseif x < y
disp('x меньше y');
else
disp('x равно y');
end
% x меньше y
```

В табл. 5.1 представлены варианты простых логических выражений оператора **if**.

Таблица 5.1

Простые логические выражения

if x < y	Истинно, если переменная x меньше переменной y и ложно в противном случае.
if x > y	Истинно, если переменная x больше переменной y и ложно в противном случае.
if x == y	Истинно, если переменная x равна переменной y и ложно в противном случае.
if x <= y	Истинно, если переменная x меньше либо равна переменной y и ложно в противном случае.
if x >= y	Истинно, если переменная x больше либо равна переменной y и ложно в противном случае.
if x ~= y	Истинно, если переменная x не равна переменной y и ложно в противном случае.

Оператор с множественным выбором **switch** <переменная> **case** <значение переменной> <операторы1> **otherwise** < операторы2> **end**. Условный

оператор **switch** осуществляет проверку на равенство некоторой переменной или вычисленного выражения, стоящего после ключевого слова **switch**, с константными значениями групп **case**.

switch

<выражение> % скаляр или строка

case <значение 1>

< операторы 1>

case <значение 2>

< операторы 2>

.....

case <значение n>

< операторы n>

otherwise

< операторы >

end

```
x=4;
y=5;
operation='*';
switch operation
case '+'
z=x+y
case '-'
z=x-y
case '*'
z=x*y
case '/'
z=x/y
case '^'
z=x^y
otherwise
disp('Введена неопределенная арифметическая операция')
end
% z = 20
```

§5.2. Циклы в MATLAB

MATLAB включает два оператора цикла: **while** и **for**.

Цикл в программе, организованный с помощью оператора **while**, в самом простом случае имеет следующий синтаксис:

while <логическое условие> <операторы> **end**

Здесь <логическое условие> подобно тому, которое применяется в операторе **if**. Цикл **while** работает до тех пор, пока <логическое условие> истинно. Если до начала выполнения цикла <логическое условие> окажется ложным, то группа инструкций <операторы>, входящих в цикл, не будут выполнены ни разу.

При необходимости значения счетчика, имеющего начальное, конечное значения и заданный шаг организация цикла в MATLAB осуществляется оператором цикла **for**, синтаксис которого имеет следующий вид:

for <счетчик> = <начальное значение>:<шаг>:<конечное значение>
<операторы цикла> **end**

Пример вычисления суммы чисел 1, 2, 3, 4, 5:

```
x=[1 2 3 4 5];  
s=sum(x); % вычисление суммы чисел функцией sum  
%s=15;  
s=0; % вычисление суммы чисел оператором цикла for  
for i=1:1:5  
    s=s+x(i);  
end  
%s=15;  
i=1; %вычисление суммы чисел оператором %цикла while  
s=0;  
while i<=5  
s=s+x(i);  
i=i+1;  
end  
%s=15
```

Контрольные вопросы

27. Какие условные операторы используются в MATLAB?
28. Опишите конструкции условных операторов, используемых в MATLAB.
29. Какие операторы циклов используются в MATLAB?
30. Опишите конструкции операторов циклов, используемых в MATLAB.

ГЛАВА 6. ПОСТРОЕНИЕ ГРАФИКОВ И ДИАГРАММ

§6.1. Двумерная графика в MATLAB

Отображение двумерных графиков на экран в MATLAB осуществляется функцией **plot()**. Функция **plot** с дополнительными параметрами записывается следующим образом:

plot(<х>, <у>, <'цвет линии, тип линии, маркер точек'>).

Если необходимо построить графики в одном окне с двумя независимыми осями, используется функция **plotyy()**, которая записывается следующим образом:

plotyy (<х1>, <у1>, <х2>, <у2>).

Отображение нескольких графиков в одном графическом окне в MATLAB осуществляется функцией **subplot()**, которая имеет несколько конструкций:

- **subplot** - создает новые объекты класса **axes** (подокна);
- **subplot(m,n,p)** - разбивает графическое окно на несколько подокон. m - число подокон по горизонтали, n - число подокон по вертикали, p - номер подокна, в которое будет выводиться текущий график (подокна отсчитываются последовательно по строкам);
- **subplot(h)**, где h - дескриптор для объекта **axes**, дает альтернативный способ задания подокна для текущего графика;
- **subplot('position',[left bottom width height])** - создает подокно с заданными нормализованными координатами (в пределах от 0.0 до 1.0);
- **subplot()** - удаляют все подокна и возвращают графическое окно в обычное состояние.

Пример

```
x=-6:0.1:6;  
subplot(2,2,1),plot(x,cos(x))  
subplot(2,2,2),plot(cos(4*x),sin(5*x+0.4))  
subplot(2,2,3),contour(peaks)%рисует контурную линию  
subplot(2,2,4),surf(peaks)%рисует непрерывную поверхность
```

На рис. 6.1 отображены результаты работы, приведенной выше программы.

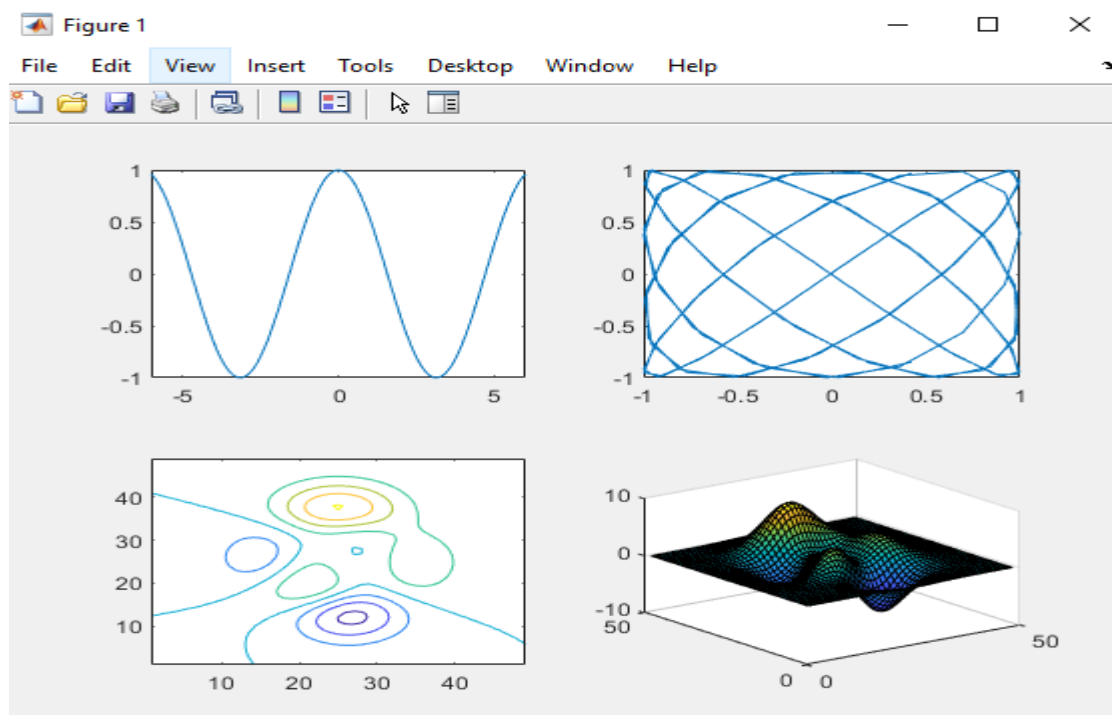


Рисунок 6.1- Графики четырех функций

Оформление графиков в MATLAB осуществляется с использованием трех параметров: цвет линии, тип линии, маркер точек (см. табл. 6.1- 6.3).

Таблица 6.1 Обозначение цвета линии графика

Маркер	Цвет линии графика
c	голубой
m	фиолетовый
y	желтый
r	красный
g	зеленый
b	синий
w	белый
k	черный

Таблица 6.2 Обозначение типа линии графика

Маркер	Тип линии графика
-	непрерывная
--	штриховая
:	пунктирная
-.	штрих-пунктирная

Таблица 6.3 Обозначение типа точек графика

Маркер	Тип точек графика
.	точка
+	плюс
*	звездочка
o	кружок
x	крестик

Для создания подписей графиков, осей и отображения сетки на графике в MATLAB используется ряд функций, приведенных в табл. 6.4.

Таблица 6.4 Функции оформления графиков

Название	Описание
grid [on, off]	включает/выключает сетку на графике
title ('заголовок графика')	создает надпись заголовка графика
xlabel ('подпись оси Oх')	создает подпись оси Oх
ylabel ('подпись оси Oу')	создает подпись оси Oу
text (x,y,'текст')	создает текстовую надпись в координатах (x,y)

Рассмотрим работу данных функций в следующем примере:

```
x = 0:0.1:2*pi;
y = cos(x);
plot(x,y,'m-o');
axis([0 1*pi -1 1]);%установка пределов по осям
grid on;
title('График функции cos(x)');
xlabel('Координата Oх');
ylabel('Координата Oу');
text(1.5,0.18,'\leftarrow cos(x)');
```

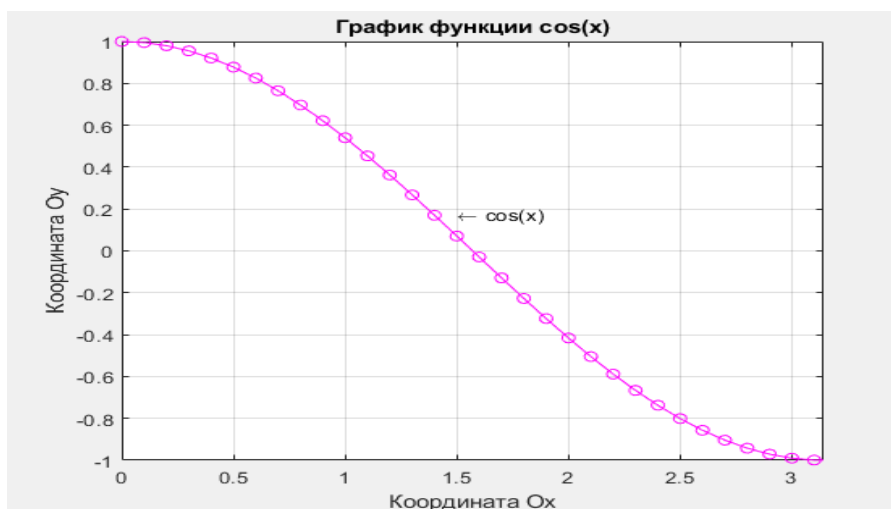


Рисунок 6.2- График функции cos(x)

§6.2. Трехмерная графика в MATLAB

Отображение графика в трехмерном пространстве в виде набора линий осуществляется функцией **plot3(X,Y,Z)**.

Пример программы отображения функции $z(x,y) = \exp(-x^2 - y^2)$ при $x = -1, -0.9, \dots, 1$; $y = -2, -1.9, \dots, 2$.

```
x=-1:0.1:1; %координаты точек по оси Ox
y=-2:0.1:2; %координаты точек по оси Oy
[X,Y]=meshgrid(x,y); %матрицы координат точек по осям
%Ox и Oy
Z=exp(-X.^2-Y.^2);
plot3(X,Y,Z);
```

Ниже на рис. 6.3 приведен результат работы программы

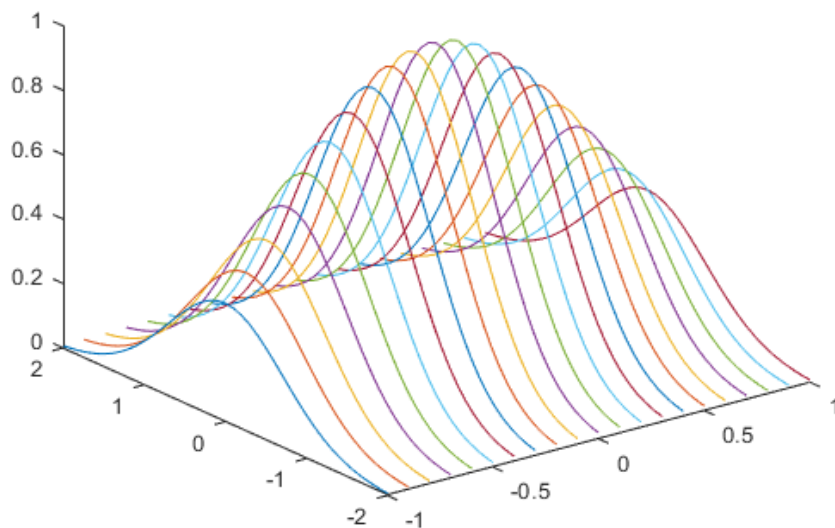


Рисунок 6.3- График функции $z(x,y) = \exp(-x^2 - y^2)$ в виде набора линий

Отображение графика в трехмерном пространстве в виде сетки осуществляется функцией **mesh(X,Y,Z)**.

```
x=-1:0.1:1; %координаты точек по оси Ox
y=-2:0.1:2; %координаты точек по оси Oy
[X,Y]=meshgrid(x,y); %матрицы координат точек по
%осям Ox и Oy
Z=exp(-X.^2-Y.^2);
mesh(X,Y,Z);
```

Ниже на рис. 6.4 приведен результат работы программы

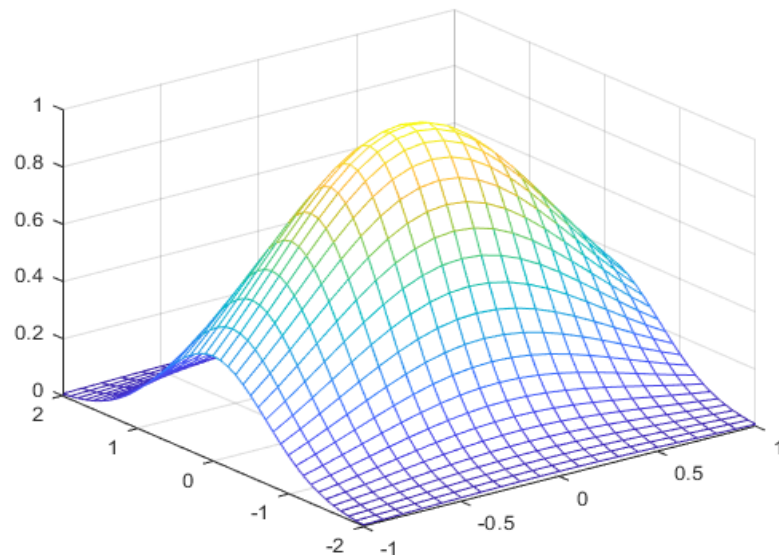


Рисунок 6.4 - График функции $z(x,y) = \exp(-x^2 - y^2)$ в виде сетки

Отображение графика в трехмерном пространстве в виде непрерывной поверхности осуществляется функцией **surf(X,Y,Z)**.

```
x=-1:0.1:1; %координаты точек по оси Ox
y=-2:0.1:2; %координаты точек по оси Oy
[X,Y]=meshgrid(x,y); %матрицы координат точек по
                     %осям Ox и Oy
Z=exp(-X.^2-Y.^2);
hidden off; %скрытые линии рисуются
surf(X,Y,Z); %отображение непрерывной поверхности
```

Ниже на рис. 6.5 приведен результат работы программы

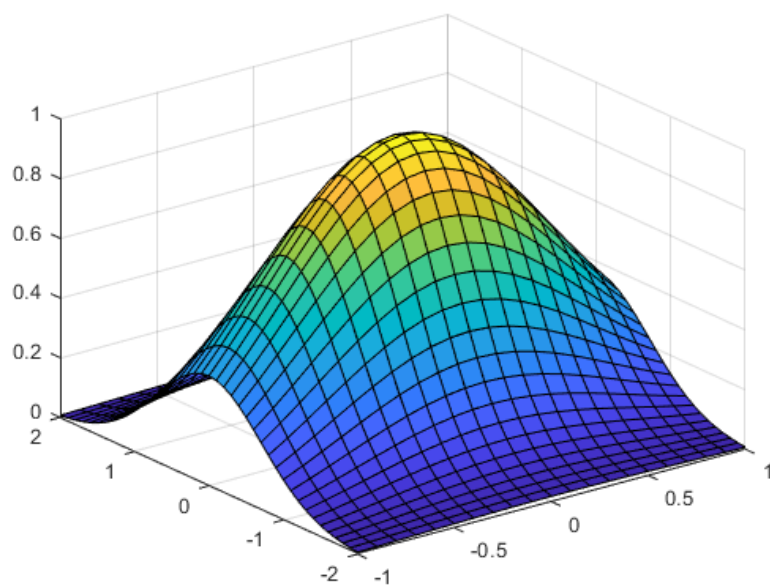


Рисунок 6.5- График функции в виде непрерывной поверхности

§6.3. Построение диаграмм и гистограмм в MATLAB

Столбчатые диаграммы без промежутков в MATLAB отображаются двумя функциями: **bar(x, Y, q, ' ')**- отображение вертикальных столбцов, **barh(...)**- отображение горизонтальных столбцов, где q- ширина столбцов, ' ' - цвет диаграммы.

Пример. Отображение функции $x(t) = \cos(0,5t) \cdot \exp(0,5t)$ на отрезке $[-2, 2]$ в виде столбчатой диаграммы без промежутков

```
t=[-2:0.1:2];  
x=cos(0.5*t).*exp(0.5*t);  
bar(t,x,1.0,'m')
```

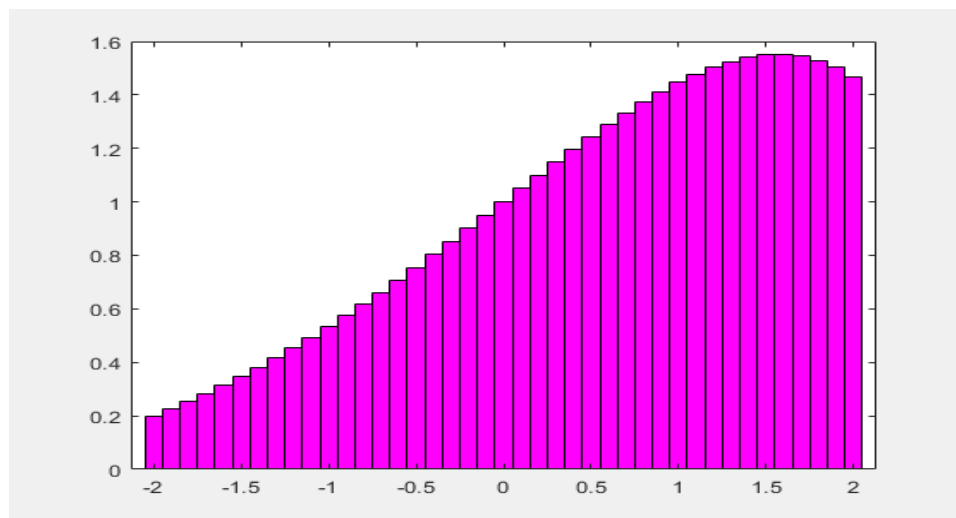


Рисунок 6.6- Столбчатая диаграмма вертикальная

```
t=[-2:0.1:2];  
x=cos(0.5*t).*exp(0.5*t);  
barh(t,x,1.0,'m')
```

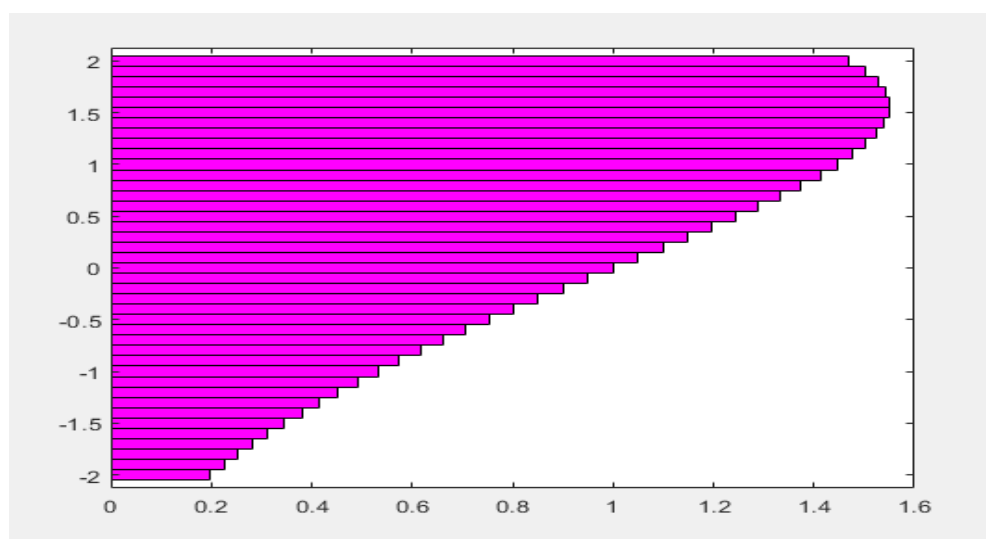


Рисунок 6.7- Столбчатая диаграмма горизонтальная

Построение круговых диаграмм осуществляется командой **pie**:

- **pie(X)** - строит круговую диаграмму по данным нормализованного вектора $X/SUM(X)$, $SUM(X)$ - сумма элементов вектора. Если $SUM(X) \leq 1.0$, то значения в X непосредственно определяют площадь секторов;
- **pie(X,EXPLODE)**- строит круговую диаграмму, у которой отрыв секторов от центра задается вектором EXPLODE, который должен иметь тот же размер, что и вектор данных X .

Пример.

```
X=[3 4 5 6];  
pie(X)
```

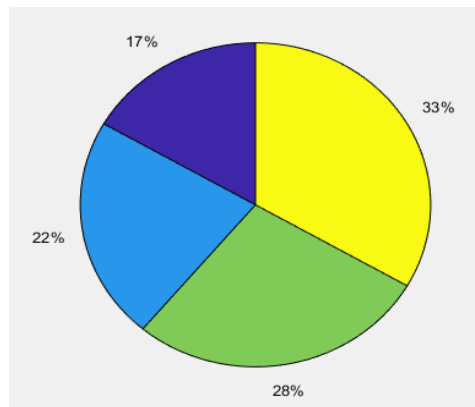


Рисунок 6.8- Круговая диаграмма без отрыва от центра

```
X=[3 4 5 6];  
pie(X,[0 0 0 2])
```

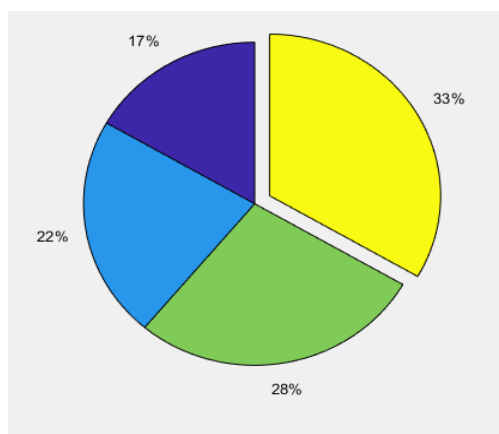


Рисунок 6.9- Круговая диаграмма с отрывом от центра

Объемные диаграммы и гистограммы в MATLAB можно получить, используя команды **bar3(..)** и **pie3(..)**.

```
t=[-2:0.1:2];  
x=cos(0.5*t).*exp(0.5*t);  
bar3(t,x,1.0,'m')
```

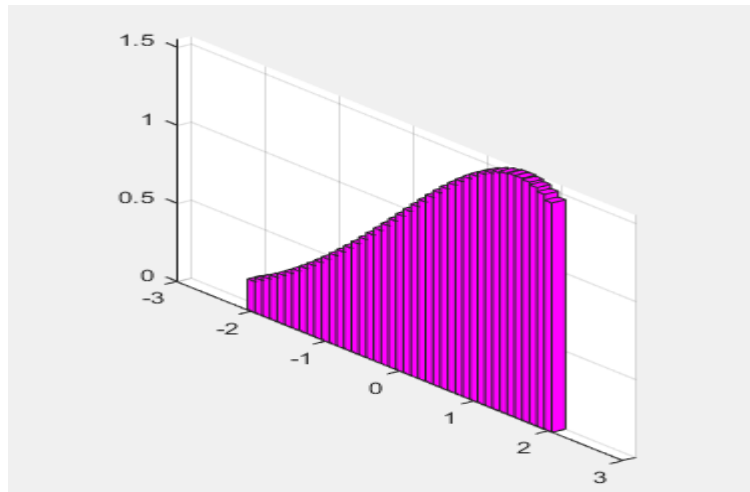


Рисунок 6.10- Объемная столбчатая диаграмма

```
X=[3 4 5 6];  
pie3(X, [0 0 0 2])
```

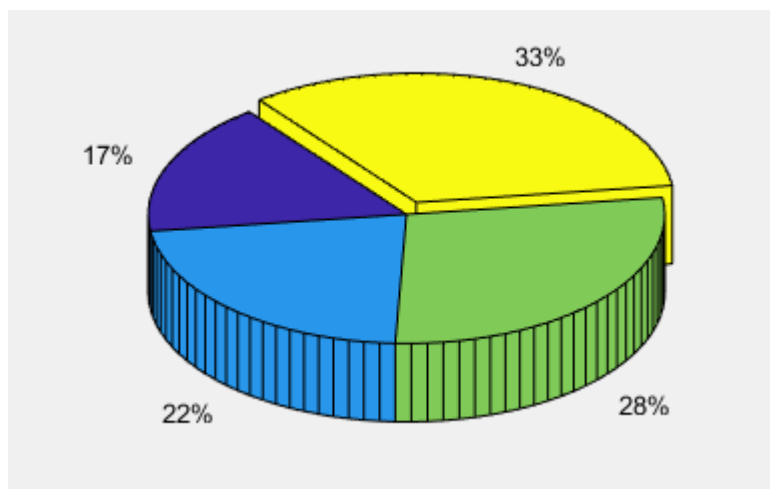


Рисунок 6.11- Объемная круговая диаграмма

§6.4. Управление световыми эффектами

Управление цветовыми палитрами в MATLAB осуществляется при помощи набора дополнительных команд, в состав которого входят:

- **col style** - выделение цвета и стиля графика из заданного массива;
- **rgbplot** - изображение палитры;
- **hsv2rgb** - преобразование палитры HSV в палитру RGB;

- **rgb2hsv** - преобразование палитры RGB в палитру HSV;
- **brighten** - управление яркостью;
- **contrast** - управление контрастом;
- **hidden** - управление показом невидимых линий;
- **whitebg** - управление цветом фона.

Контрольные вопросы

1. Какая функция используется для отображения двумерных графиков в MATLAB?
2. Какая функция используется для отображения трехмерных графиков в MATLAB?
3. Приведите дополнительные параметры функции **plot()**.
4. Какая функция используется для отображения нескольких графиков в одном окне в MATLAB?
5. Как задаются цвета линий в MATLAB?
6. Как задаются типы линий в MATLAB?
7. Как создаются подписи графиков в MATLAB?
8. Как создаются подписи осей графиков в MATLAB?
9. Как создается отображение сетки на графике в MATLAB?
10. Какие функции используются для отображения столбчатых вертикальных и горизонтальных диаграмм в MATLAB?
11. Какие функции используются для отображения круговых диаграмм в MATLAB?
12. Какая функция используется для отображения объемных столбчатых диаграмм в MATLAB?
13. Какая функция используется для отображения объемных круговых диаграмм в MATLAB?
14. Какая команда управления используется для выделения цвета и стиля графика в MATLAB?
15. Какая команда управления используется для изображения палитры в MATLAB?
16. Какая команда применяется для управления яркостью в MATLAB?

17. Какая команда применяется для управления контрастом в MATLAB?
18. Какая команда применяется для управления показом невидимых линий в MATLAB?
19. Какая команда применяется для управления цветом фона в MATLAB?

ГЛАВА 7. ПРОГРАММИРОВАНИЕ ПРОЦЕДУР И ФУНКЦИЙ, РАБОТА С ФАЙЛАМИ

§7.1. Программирование процедур и функций

Реализация повторяющихся вычислений осуществляется путем создания процедур и функций пользователя и вызова их по мере необходимости. Процедуры и функции пользователя могут быть записаны в виде m-файлов. Причем, имя m-файла должно совпадать с именем процедуры или функции. Необходимым условием так же является то, что основная программа и m-файл с процедурой или функцией должны находиться в одном и том же месте. В процедуре могут быть один или несколько выходных параметров и входных аргументов:

function [val1, val2,...] = **Procedure name** (arg1, arg2,...)
 <тело процедуры> **end**

где val1, val2, ... - выходные параметры; arg1, arg2, ... - входные аргументы;
<тело процедуры> - группа инструкций (операторов), выполняемых при вызове функции.

Обращение к процедуре выглядит следующим образом:

[val1, val2, ...] = **Procedure name** (arg1, arg2, ...)

Функция может иметь только один выходной параметр, который вычисляется в последнем исполняемом операторе функции:

function resfunc= **Function name** (arg1, arg2, ...)
 <тело функции> **end**

где resfunc- имя выходной параметра.

Обращение к функции выглядит следующим образом:

Function name (arg1, arg2, ...)

При обращении к процедурам и функциям необходимо задавать конкретные значения входных аргументов `arg1, arg2, ...`.

Примеры оформления процедуры и функции приведены ниже:

```
%Процедура
function [x1,x2] = form(a,b,c)
d = sqrt(b^2 - 4*a*c);
x1 = (-b + d) / (2*a);
x2 = (-b - d) / (2*a);
end
```

```
%Обращение к процедуре
[x1,x2] = form(1,1,1)
% x1 = -0.5000 + 0.8660i
% x2 = -0.5000 - 0.8660i
```

```
%Функция
function x = funfit(a,b,c)
x=(b+c+a)/(2*a);
end
```

```
%Обращение к функции
funfit(2,1,1)
% x = 1
```

Оператор `@` позволяет создавать анонимные функции:

```
fh=@funfit
feval(fh,2,1,1)
% x = 1
```

§7.2. Работа с файлами в MATLAB

Сохранение и последующая загрузка данных в MATLAB осуществляется функциями **save** и **load**:

save <имя файла> <имена переменных> % сохранение данных;

load <имя файла> <имена переменных> % загрузка данных.

Функция **save** производит сохранение произвольных переменных программы в файл, который по умолчанию располагается в рабочем каталоге (обычно поддиректория `work`) и в поздних версиях MATLAB имеет расширение `m` (в ранних версиях `mat`). Соответственно, функция **load** производит загрузку из указанного `m`-файла ранее сохраненных переменных.

```

x1 = 1;
x2 = 2;
x3 = 3;
x4 = 4;
x5 = 5;
X=[x1 x2 x3 x4 x5];%формирование массива X
save NOpt.txt X -ascii;%запись массива X в файл NOpt.txt

A = load('N35.txt');%загрузка таблицы с данными из
                    %файла N35.txt
% раскидываем столбцы
Z=A(:,1);%столбец с номерами строк таблицы
X1= A(:,2);%столбец со значениями параметра X1
X2 = A(:,3);%столбец со значениями параметра X2
X3 = A(:,4);%столбец со значениями параметра X3
X4 = A(:,5);%столбец со значениями параметра X4
x5 = A(:,6);%столбец со значениями параметра X5
N = length(Z);%определение числа строк таблицы N

```

§7.3. Специальные функции MATLAB для работы с передаточными функциями

При моделировании систем управления в MATLAB используются функции пакета прикладных программ (ППП) Control System Toolbox (CONTROL), который сосредоточен в подкаталоге CONTROL каталога TOOLBOX.

Основными вычислительными объектами этого ППП являются:

- родительский объект (класс) LTI (Linear Time-Invariant System - линейные, инвариантные во времени системы) или линейные стационарные системы (ЛСС).
- дочерние объекты (классы) - подклассы класса LTI, соответствующие разным представлениям ЛСС.

Таблица 7.1

Разные формы представления ЛСС

Тип модели	Описание
tf	Модель передаточной функции в полиномиальной форме
zpk	Модель передаточной функции в (факторизованной) форме с нулевым коэффициентом усиления
ss	Модель пространства состояний
frd	Модель данных частотной характеристики
pid	Параллельный ПИД-регулятор

Объект LTI содержит информацию, не зависящую от конкретного представления ЛСС, а также от имен входов и выходов. Дочерние объекты определяются конкретной формой представления ЛСС, т.е. зависят от модели представления. Объект класса TF характеризуется векторами коэффициентов числителя и знаменателя рациональной передаточной функции. Объект класса ZPK характеризуется векторами, содержащими значения нулей, полюсов передаточной функции системы и коэффициента передачи системы. Объект класса SS определяется четверкой матриц, описывающих динамическую систему в пространстве состояния. Объект класса FRD создается на основании отклика системы на заданном частотном спектре и ориентирован на применение при проведении измерительных экспериментов.

Виды LTI-систем:

- **SISO** (Single In Single Out) - одномерная (ОМ) система, т.е. система с одним входом и одним выходом.
- **MIMO** (Multiple Input Multiple Output) - многомерная (ММ) система с несколькими входами и выходами.
- **TF**-объект (Transfer Function - передаточная функция);
- **ZPK**-объект (Zero-Pole-Gain - нули-полюсы-коэффициент передачи);
- **SS**-объект (State Space - пространство состояния).

При создании LTI - модели и преобразовании из одной формы представления в другую используются функции **tf**, **zpk**, **ss**.

Таблица 7.2 Функции создания LTI - модели и преобразования из одной формы представления в другую

tf	Создание и преобразование в передаточные функции
zpk	Создание и преобразование в нули/полюсы/коэффициент усиления
ss	Создание и преобразование в пространство состояния

tf создает непрерывную передаточную функцию SYS с числителем NUM и знаменателем DEN: $SYS = tf(NUM, DEN)$

zpk создает модели с нулевым усилением или конвертирует в формат с

нулевым усилением: $SYS = zpK(Z, P, K)$ - модель SYS с нулевым коэффициентом усиления (ZPK) с нулями Z , полюсами P и коэффициентами усиления K .

ss создает модель в пространстве состояний или преобразует модель в пространство состояний: $SYS = ss(A, B, C, D)$ - создает объект SS SYS , представляющий модель непрерывного времени в пространстве состояний:

$$dx / dt = Ax(t) + Bu(t)$$

$$y(t) = Cx(t) + Du(t)$$

Вы можете установить $D = 0$ для обозначения нулевой матрицы соответствующих размеров.

Таблица 7.3 Таблица полей **NUM** и **DEN** функции **TF**

Специфические поля передаточных функций (TF-объектов)	
num	Числитель - вектор-строка для ОМ-систем; для ММ-систем - массив ячеек из векторов-строк размером NY на NU (например, {[1 0] 1; 3 [1 2 3]})
den	Знаменатель - вектор-строка для ОМ-систем; для ММ-систем - массив ячеек из векторов-строк размером NY на NU . Например: <code>tf([-5 ; [1 -5 6]] , {[1 -1] ; [1 1 0]})</code> определяет систему с одним входом и двумя выходами [-5 /(s-1)] [(s^2-5s+6)/(s^2+s)]
Variable	Имя (тип) переменной Возможны варианты: s, p, z По умолчанию принимается s (для непрерывных переменных) и z (для дискретных). Имя переменной влияет на отображение и создает дискретную ПФ для дискретных сигналов

Таблица 7.4 Специфические атрибуты ZPK -объектов

Специфические поля передаточных функций (ZPK -объектов)	
z	Нули Вектор-строка для ОМ-систем; для ММ-систем - массив ячеек из векторов-строк размером NY на NU
p	Полюсы Вектор-строка для ОМ-систем; для ММ-систем - массив ячеек из векторов-строк размером NY на NU
k	Коэффициенты передачи

	Число - для ОМ-систем, матрица NY на NU для ММ-систем
Variable	Имя (тип) переменной (из перечня) То же, что и для TF-объекта (см. выше)

Таблица 7.5 Специфические атрибуты SS-объектов (моделей пространства состояния)

Специфические поля передаточных функций (SS -объектов)	
a,b,c,d	A,B,C,D - матрицы, в соответствии с уравнениями в переменных состояния: $x = Ax + Bu, y = Cx + Du$
e	E - матрица для систем Descriptor'a (описателя). По умолчанию $E = eye(size(A))$
StateName	Имя переменной состояния (не обязательное). Массив ячеек NX на I из строк (используйте " для состояний без имени). Пример: {'положение'; 'скорость'}

Ниже приведены общие для всех LTI-моделей атрибуты и примеры работы с передаточными функциями с использованием пакета CONTROL.

Таблица 7.6 Атрибуты, общие для всех LTI-моделей

Ts	Дискрет по времени (в секундах) Положительный скаляр (период дискретизации) для дискретных систем. $Ts = -1$ для дискретных систем с неустановленной частотой дискретизации. $Ts = 0$ для непрерывных систем.
Td	Задержки входов (в секундах) Вектор 1 на NU промежутков времен задержек входов. Установка Td как скаляра определяет единую задержку по всем входам. Используется только для непрерывных систем. Используйте D2D для установки задержек в дискретных системах. $Td = []$ для дискретных систем
InputName	Имена входов Строка для систем с одним входом. Массив ячеек NU на 1 из строк для систем с несколькими входами (используйте " для переменных без имени). Примеры: 'момент' или {'напор1'; 'отклонение элеронов1'}
OutputName	Имена выходов Строка для систем с одним выходом. Массив ячеек NY на 1 из строк для систем с несколькими выходами (используйте " для переменных без имени). Пример: 'мощность' или {'скорость'; 'угол атаки'}
Notes	Заметки

	Любая строка или массив ячеек из строк символов. Пример: 'Эта модель создана в январе 2000'
Userdata	Дополнительная информация или данные. Может быть любого типа MATLAB

```

w1 = tf([3 1], [1 1 1], 'variable', 's')
w2 = tf([3 1], [1 1 1], 0.1, 'variable', 'z')
w3 = tf([3 1], [1 1 1], 'variable', 'p')
w1 =
      3 s + 1
      -----
      s^2 + s + 1
w2 =
      3 z + 1
      -----
      z^2 + z + 1
Sample time: 0.1 seconds
Discrete-time transfer function.
w3 =
      3 p + 1
      -----
      p^2 + p + 1
w1 = tf([3 1], [1 1 1])
zpk(w1)
ans =
      3 (s+0.3333)
      -----
      (s^2 + s + 1)
w1 = tf([3 1], [1 1 1]);
ss(w1)
ans =
  A =
      x1  x2
x1  -1  -1
x2   1   0
  B =
      u1
x1   2
x2   0
  C =
      x1  x2
y1  1.5  0.5
  D =
      u1
y1   0

```

Преобразование моделей в MATLAB осуществляется функциями, приведенными в таблице 7.7.

Таблица 7.7 Функции преобразования моделей

parallel	параллельное соединение
feedback	соединение с обратной связью
series	последовательное соединение
append, connect	различные соединения блоков

Пример

```
w1 = tf([3 1], [1 1 1])
w2 = tf([4 1], [2 1 1])
w3 = tf([5 1], [3 1 1])
sys1=parallel(w1,w2)%параллельное соединение
sys2=series(w1,w2)%последовательное соединение
sys3=feedback(sys2,w3)%с отрицательной обратной связью
sys5=feedback(sys2,w3,+1)%положительной обратной связью
w1 =
```

$$\frac{3s + 1}{s^2 + s + 1}$$

$$w2 =$$

$$\frac{4s + 1}{2s^2 + s + 1}$$

$$w3 =$$

$$\frac{5s + 1}{3s^2 + s + 1}$$

$$sys1 =$$

$$\frac{10s^3 + 10s^2 + 9s + 2}{2s^4 + 3s^3 + 4s^2 + 2s + 1}$$

$$sys2 =$$

$$\frac{12s^2 + 7s + 1}{2s^4 + 3s^3 + 4s^2 + 2s + 1}$$

$$sys3 =$$

$$\frac{36s^4 + 33s^3 + 22s^2 + 8s + 1}{6s^6 + 11s^5 + 17s^4 + 73s^3 + 56s^2 + 15s + 2}$$

$$sys5 =$$

$$\frac{36s^4 + 33s^3 + 22s^2 + 8s + 1}{6s^6 + 11s^5 + 17s^4 - 47s^3 - 38s^2 - 9s}$$

Функция **step** осуществляет построение переходных процессов для сформированных передаточных функций в графическом окне.

Таблица 7.8

Формы записи функции step

<u>step(sys)</u>	отображает реакцию модели динамической системы на входной шаг с единичной амплитудой
<u>step(sys,tFinal)</u>	моделирует переходную характеристику от $t = 0$ до конечного момента времени $t = tFinal$.
<u>step(sys,t)</u>	строит переходную характеристику в моменты времени, которые вы указываете в векторе t.
<u>step(sys1,sys2,...,sysN)</u>	строит на одном графике переходную характеристику нескольких динамических систем. Все системы должны иметь одинаковое количество входов и выходов.
<u>step(sys1,LineSpec1,...,sysN, LineSpecN)</u>	задает цвет, стиль линии и маркер для каждой системы на графике. Вы можете использовать LineSpec с любой из предыдущих комбинаций входных аргументов. Если вам нужны дополнительные параметры настройки графика, используйте вместо них stepplot.
<u>step(____, opts)</u>	задает дополнительные параметры для вычисления реакции на скачок, такие как амплитуда шага или входное смещение. Используйте stepDataOptions, чтобы создать opts набора параметров.

Пример использования функции step

```
w1 = tf([3 1], [1 1 1]);
step(w1)
w1 =
      3 s + 1
-----
    s^2 + s + 1
```

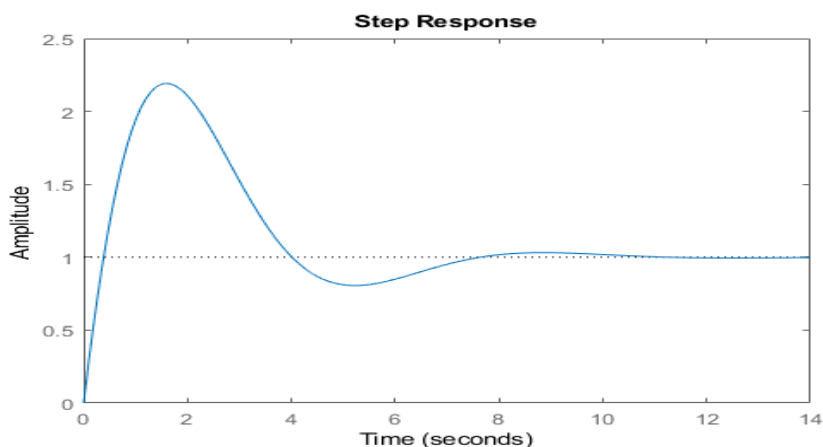


Рисунок 7.1- График переходного процесса

Функция **dcgain** осуществляет определение статических коэффициентов передачи сформированных передаточных функций.

Пример

```
w1 = tf([3 1], [1 1 1]);
dcgain(w1)
ans =
    1
```

Функция **damp** вычисляет показатели демпфирования и собственные частоты системы.

Пример

```
% исходные данные
H = 2000; Jp = 0.8;
Jst = 125; Dst = 300; Dp = 0;
Kr = 50000;
% расчет параметров
Omega = H * sqrt(1+Dst*Dp/H^2)/sqrt(Jst * Jp);
t0 = 1/Omega;
dzeta =
((Dp/(2*H))/sqrt(Jst/Jp)+(Dst/(2*H))*sqrt(Jp/Jst))*...
(1/sqrt(1+(Dst*Dp)/H^2));
t1 = 2*t0;
% формирование передаточных функций
w1 = tf([Kr/H],[1 0]);
w2 = tf([1],[t0^2 2*t0*dzeta 1]);
w3 = tf([1],[t1 1])*tf([1],[t1 1]); % корректирующее
звено
% передаточная функция гиростабилизатора с коррекцией
W = w1 * w2 * w3;
damp(W)
Pole                Damping                Frequency                Time Constant
(rad/seconds)                (seconds)
0.00e+00                -1.00e+00                0.00e+00
-1.00e+02 + 1.40e-06i    1.00e+00                1.00e+02                1.00e
-1.00e+02 - 1.40e-06i    1.00e+00                1.00e+02                1.00e
-1.20e+00 + 2.00e+02i    6.00e-03                2.00e+02                8.33e
-1.20e+00 - 2.00e+02i    6.00e-03                2.00e+02                8.33e
```

Функция **eig(pole)** осуществляет определение корней сформированных передаточных функций.

Пример

```
w1 = tf([3 1], [1 1 1]);
w2 = tf([3 1], [1 2 1]);
eig(w1)
pole(w2)

ans =
```

```

-0.5000 + 0.8660i
-0.5000 - 0.8660i
ans =
-1
-1

```

Функция **tzero** осуществляет определение полюса сформированных передаточных функций.

Пример

```

w1 = tf([3 1], [1 1 1]);
tzero(w1)
ans =
-0.3333

```

Функция **pzmap** позволяет рассмотреть нули и полюса сформированных передаточных функций.

Пример

```

w1 = tf([3 1], [1 1 1]);
pzmap(w1)

```

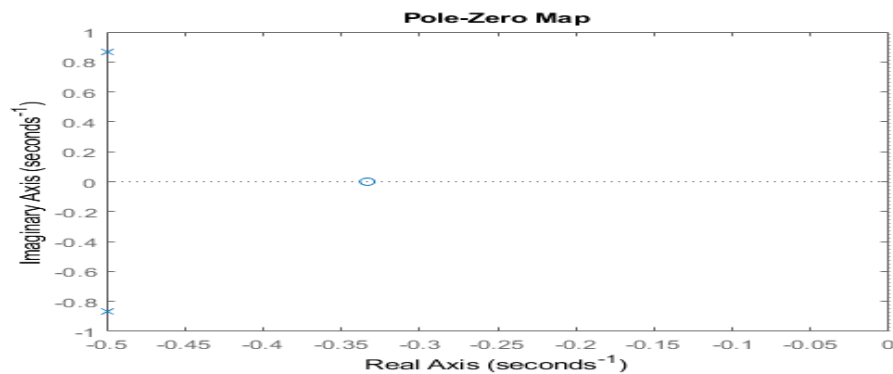


Рисунок 7.2- Нули и полюса передаточной функции

Функция **pzmap** в сочетании с **grid** осуществляет вычисление и вывод нулей и полюсов на графике по сетке.

Пример

```

w2=tf([3 1], [1 4 2])
pzmap(w2)
grid
w2 =
      3 s + 1
-----
s^2 + 4 s + 2

```

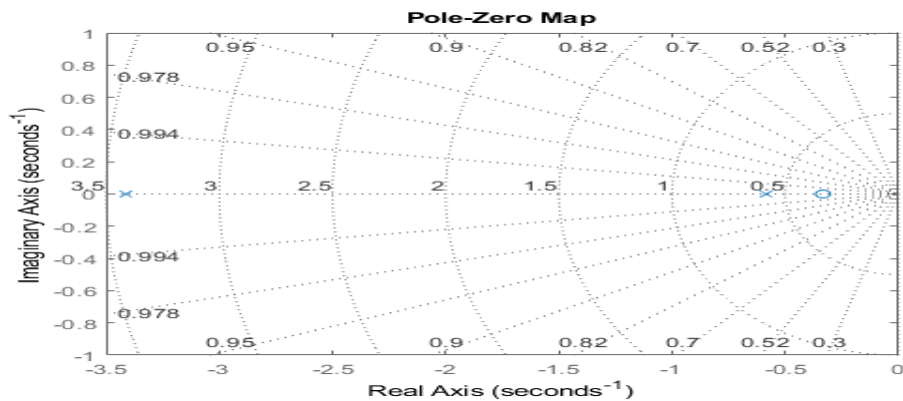


Рисунок 7.3- График нулей и полюсов передаточной функции

При помощи функции **pzmap** можно получить значения полюсов и нулей системы:

```
w=tf([3 1], [1 4 2])
[np,dz]=pzmap(w)
w =
      3 s + 1
-----
s^2 + 4 s + 2
np =
    -3.4142
    -0.5858
dz =
    -0.3333
0
```

Контрольные вопросы

1. Какой вид имеет определение собственных функций в MATLAB?
2. Какие функции используются для сохранения и загрузки данных в MATLAB?
3. Какие основные вычислительные объекты входят в пакет прикладных программ Control System Toolbox?
4. Чем характеризуется объект класса TF в MATLAB?
5. Чем характеризуется объект класса ZPK в MATLAB?
6. Чем определяется объект класса SS в MATLAB?
7. На основании чего создается и на что ориентирован объект класса FRD?
8. Какие функции для создания LTI - модели и преобразования из одной формы представления в другую используются в MATLAB?
9. Какие поля NUM и DEN имеет функция TF в MATLAB?

10. Какие поля передаточных функций ZPK -объектов имеются в MATLAB?
11. Какие поля передаточных функций SS -объектов имеются в MATLAB?
12. Какие атрибуты, общие для всех LTI-моделей, имеются в MATLAB?
13. Какими функциями осуществляется преобразование моделей в MATLAB?
14. Что осуществляет функция step в MATLAB?
15. Что осуществляет функция dsgain в MATLAB?
16. Что осуществляет функция eig(pole) в MATLAB?
17. Что осуществляет функция tzero в MATLAB?
18. Что осуществляет функция pzmap в сочетании с grid в MATLAB?

ЧАСТЬ II. ОСНОВЫ РАБОТЫ С ДАННЫМИ И МОДЕЛЯМИ

ГЛАВА 8. ФУНКЦИИ ИЗВЛЕЧЕНИЯ ДАННЫХ ИЗ МОДЕЛИ И АНАЛИЗА СВОЙСТВ МОДЕЛИ

§8.1. Функции извлечения данных из модели

Функции извлечения данных из модели приведены в таблице 8.1.

Таблица 8.1 Функции извлечения данных из модели

tfdata	Извлечение параметров передаточных функций
zpkdata	Извлечение нулей/полюсов/коэффициентов усиления
ssdata	Извлечение матриц пространства состояния
get	Получение свойств LTI- модели

Пример

```
w1 = tf([3 1], [1 1 1]);
tfdata(w1)
w2=zpk(w1)
zpkdata(w2)
w3=ss(w1)
ssdata(w3)
get(w1)
ans =
    1×1 cell array
        {[0 3 1]}
w2 =
    3 (s+0.3333)
    -----
    (s^2 + s + 1)
Continuous-time zero/pole/gain model.
```

```

ans =
    1×1 cell array
        {[-0.3333]}
w3 =
    A =
        x1  x2
    x1  -1  -1
    x2   1   0
    B =
        u1
    x1   2
    x2   0
    C =
        x1  x2
    y1  1.5  0.5
    D =
        u1
    y1   0
Continuous-time state-space model.
ans =

    -1    -1
     1     0
    Numerator: {[0 3 1]}
    Denominator: {[1 1 1]}
    Variable: 's'
    IODelay: 0
    InputDelay: 0
    OutputDelay: 0
    Ts: 0
    TimeUnit: 'seconds'
    InputName: {''}
    InputUnit: {''}
    InputGroup: [1×1 struct]
    OutputName: {''}
    OutputUnit: {''}
    OutputGroup: [1×1 struct]
    Notes: [0×1 string]
    UserData: []
    Name: ''
    SamplingGrid: [1×1 struct]

```

При обращении к функции **tfdata** без указания параметров выводится информация только о структуре данных модели. Если необходимо получить коэффициенты полиномов в векторной форме, следует задать параметр 'v': `tfdata(SYS,'v')`. Задав в качестве выходных параметров два вектора **n** и **d**: `[n,d] = tfdata(SYS,'v')`, можно получить данные числителя и знаменателя.

Проверка результатов вычислений и преобразований моделей по корням числителя и знаменателя передаточных функций может быть осуществлена с использованием функции **roots**.

Пример

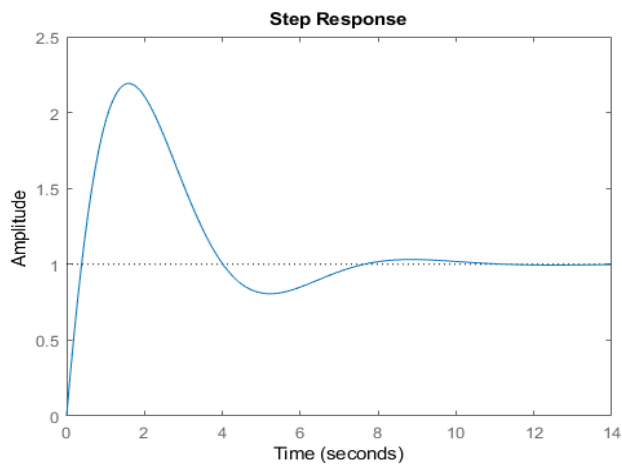
```
w1 =tf([3 1], [1 1 1]);  
roots([3 1])  
roots([1 1 1])  
ans =  
    -0.3333  
ans =  
    -0.5000 + 0.8660i  
    -0.5000 - 0.8660i
```

§8.2. Функции анализа свойств модели

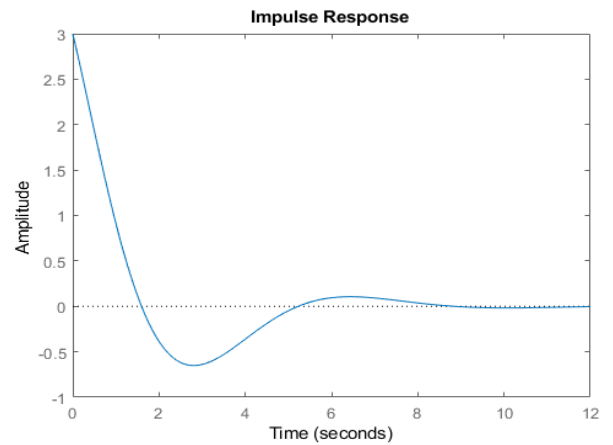
Анализа свойств модели в MATLAB может осуществляться функциями **step** (отображает реакцию модели динамической системы на ступенчатое возмущение), **impulse** (отображает реакцию модели динамической системы на импульсное возмущение), **bode** (отображает графики амплитуды и фазы частотной характеристики динамической системы), **margin** (отображает реакцию Бode динамической системы и указывает на графике запасы усиления и фазы), **rlocus** (отображает корневой годограф модели SISO динамической системы).

Пример

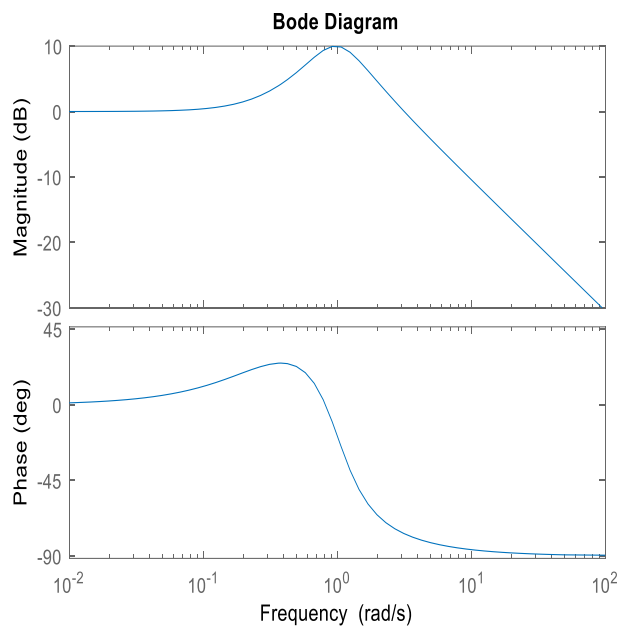
```
w1 =tf([3 1], [1 1 1]);  
step(w1)  
impulse(w1)  
bode(w1)  
margin(w1)  
rlocus(w1);
```



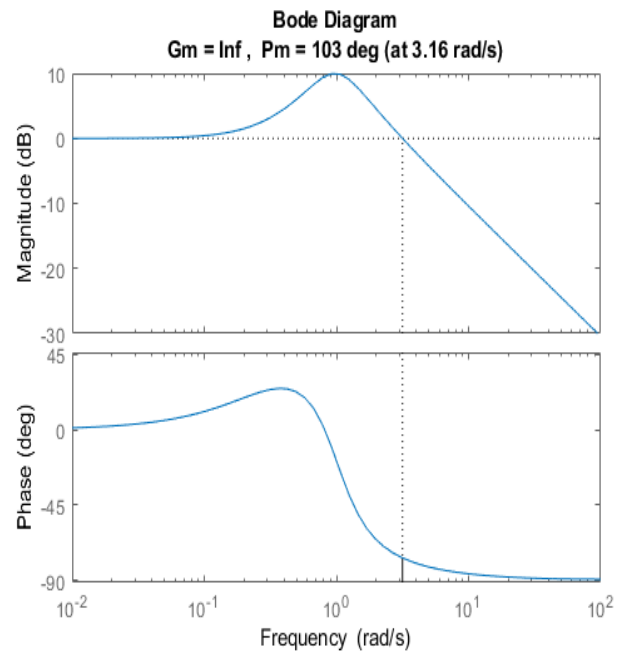
a



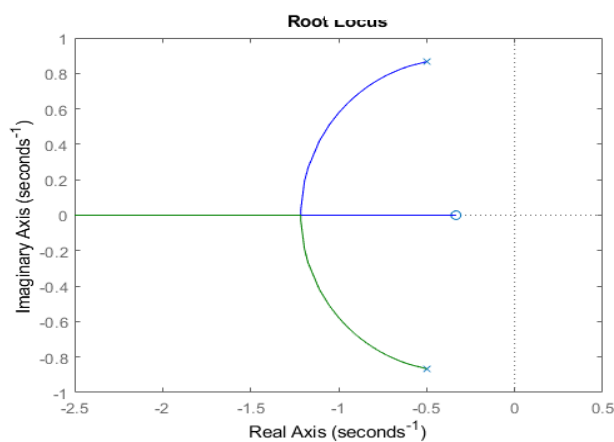
б



в



г



д

Рисунок 8.1- Графики анализа модели функциями: *a*- step; *б*- impulse; *в*- bode; *г* - margin; *д*- rlocus

Функция **damp** осуществляет вычисление показателей демпфирования и собственных частот системы:

```
w=tf([25], [2.5e-009 5.06e-007 0.0001262 0.02006 1 0])
damp(w);
w =
```

25

2.5e-09s^5+5.06e-07s^4+0.0001262s^3+0.02006s^2+s

Eigenvalue	Damping	Freq. (rad/s)
0.00e+000	-1.99e+000	0.00e+000
-1.00e+002	1.00e+000	1.00e+002
-1.00e+002	1.00e+000	1.00e+002
-1.20e+000 +2.00e+002i	6.00e-003	2.00e+002
-1.20e+000 -2.00e+002i	6.00e-003	2.00e+002

Функция **stepinfo** осуществляет вычисление характерных точек графика.

Пример

```
w1=tf([9], [1 2.4 9])
s1 = stepinfo(w1);
w1 =
```

9

s^2 + 2.4 s + 9

s1 =

struct with fields:

```

    RiseTime: 0.4884
    SettlingTime: 2.8031
    SettlingMin: 0.9065
    SettlingMax: 1.2537
    Overshoot: 25.3741
    Undershoot: 0
         Peak: 1.2537
    PeakTime: 1.1513
```

Контрольные вопросы

1. Какие функции извлечения данных из модели существуют в MATLAB?
2. Какая функция используется для извлечения параметров передаточных функций в MATLAB?
3. Какая функция используется для извлечения нулей/ полюсов/ коэффициентов усиления в MATLAB?
4. Какая функция используется для извлечения матриц пространства состояния в MATLAB?
5. Какая функция используется для в получения свойств LTI- модели

MATLAB?

6. Какие функции анализа свойств модели используется в MATLAB?
7. Какие показатели системы вычисляет функция `damp` в MATLAB?
8. Какие показатели вычисляет функция `stepinfo` в MATLAB?

ГЛАВА 9. LTIVIEWER - ИНТЕРАКТИВНЫЙ ОБОЗРЕВАТЕЛЬ СВОЙСТВ ЛИНЕЙНЫХ МОДЕЛЕЙ

§9.1. Задание идентификаторов созданных LTI - систем во входных параметрах

Функция **ltiview** осуществляет задание идентификаторов созданных LTI - систем во входных параметрах.

Пример

```
w=tf([1],[1 1]);  
w1=tf([1 1],[1 1 1]);  
ltiview(w,w1)
```

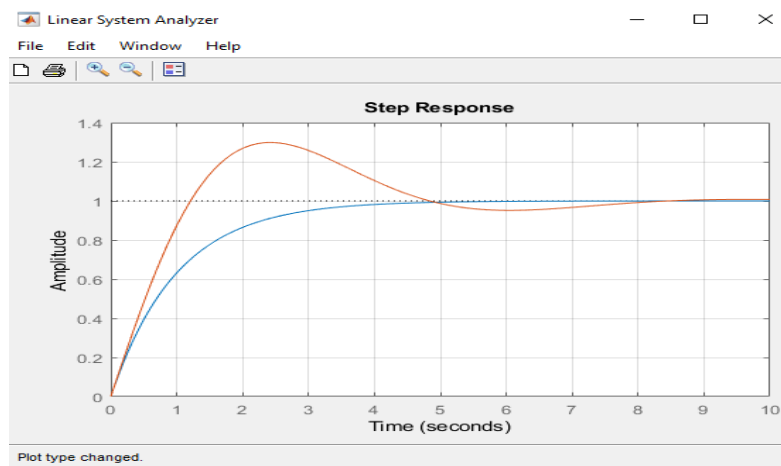


Рисунок 9.1- Графики созданных моделей

При нажатии на правую кнопку мыши открывается выпадающее меню, которое позволяет выполнить настройку графиков, выбрать необходимую характеристику системы (реакция на ступенчатый или импульсный входной сигнал, ЛАФЧХ и др.) и другие параметры обозревателя. Задание параметров осуществляется в выпадающем меню **Properties**. На рисунке 9.1 приведены различные характеристики систем.

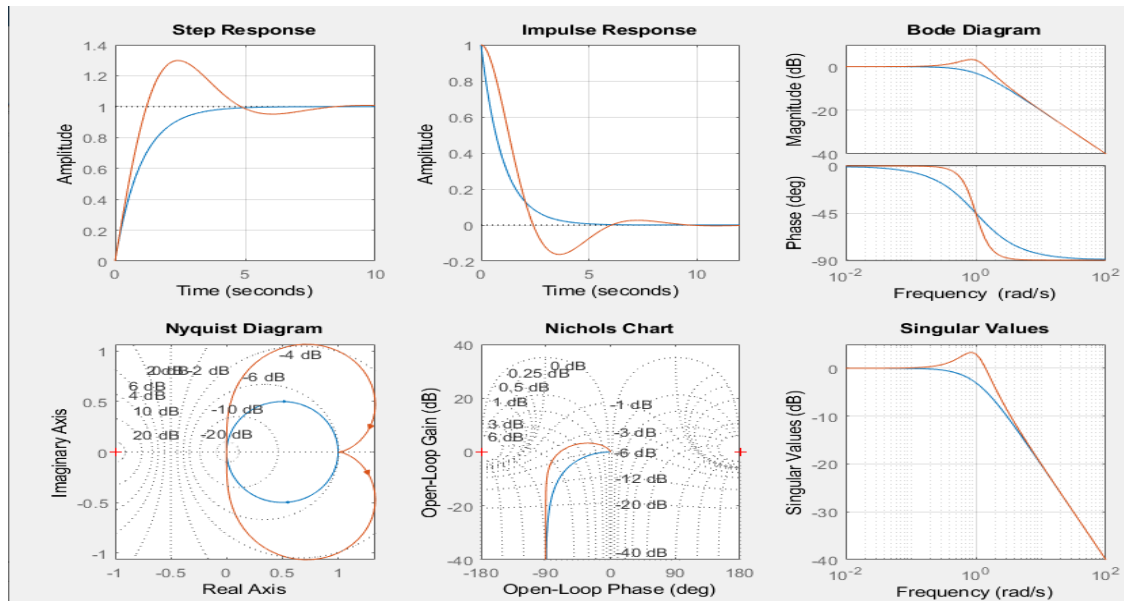


Рисунок 9.2- Характеристики систем

a - реакция на ступенчатое возмущение; *б* - реакция на импульсное возмущение; *в* - диаграмма Боде; *г* - диаграмма Найквиста; *д* - диаграмма Николса; *е* - особые значения

§9.2. Представление LTI-системы в пространстве состояний

SS Модель «Вход - Выход» в пространстве состояний в MATLAB формируется на основе системы дифференциальных уравнений первого порядка, разрешенных относительно производных

$$\begin{cases} \dot{X} = AX + BU \\ \dot{Y} = CX + DU \end{cases}, \quad (9.1)$$

получившей название нормальной формой Коши.

Создание модели динамической системы в пространстве состояния в MATLAB производится функцией `ss` из пакета Control System Toolbox.

Параметрами модели являются элементы матриц A , B , C , D .

Матрицы, входящие в модель, имеют следующую размерность:

$X(t)$ - вектор состояния-столбец, который имеет размерность n и включает переменные объекта, однозначно задающие его состояние; A - квадратная матрица размерностью $[n \times n]$ параметров системы; B - матрица управления размерностью $[n \times m]$, в которой i -й столбец ($1 \leq i \leq m$) содержит параметры, характеризующие воздействие i -го входного сигнала на соответствующий j -й параметр состояния ($1 \leq j \leq n$); C - матрица выхода размерностью $[n \times r]$, ($k \leq n$);

D - матрица прямой связи; n - число переменных состояния; m - количество входных сигналов (сигналы управления), r - количество выходных сигналов.

Пример

Заданы уравнения объекта в форме Коши, описывающие короткопериодическое продольное движение летательного аппарата:

$$\begin{cases} \frac{d\omega_z}{dt} = (-c_1 - c_5)\omega_z + (-c_4c_5 - c_2)\alpha - c_3\delta_B \\ \frac{d\alpha}{dt} = \omega_z - c_4\alpha \end{cases}$$

Система обыкновенных дифференциальных уравнений включает: два выходных параметра ω_z - угловая скорость тангажа и α - угол атаки; δ_B - угол отклонения высоты.

Для работы с функциями MATLAB представим дифференциальные уравнения в матричной форме:

$$X - \text{вектор состояния: } X = \begin{bmatrix} \omega_z \\ \alpha \end{bmatrix}$$

$$A - \text{матрица коэффициентов: } A = \begin{bmatrix} (-c_1 - c_5) & (-c_4c_5 - c_2) \\ 1 & -c_4 \end{bmatrix}$$

$$\begin{cases} \dot{X} = AX + BU \\ \dot{Y} = CX + DU \end{cases} \quad U - \text{вектор входа системы: } U = \begin{bmatrix} \delta_B \\ 0 \end{bmatrix}$$

$$B - \text{матрица управления: } B = \begin{bmatrix} -c_3 \\ 0 \end{bmatrix}$$

$$C - \text{матрица выхода: } C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

D - матрица, характеризующая

$$\text{связь входного сигнала с выходным: } D = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Y - вектор выхода системы.

```
% ss system
c1=6.33;c2=16.543;c3=12.317;c4=0.973;c5=0.453;
A = [(-c1 -c5) (c4*c5 -c2)
      1 -c4];
B = [c3
      0]; c = [1 0
               0 1]; d = 0;
Model_ss = ss(A,B,c,d,'statename',{'Om_z' 'alfa'},...
```

```



```

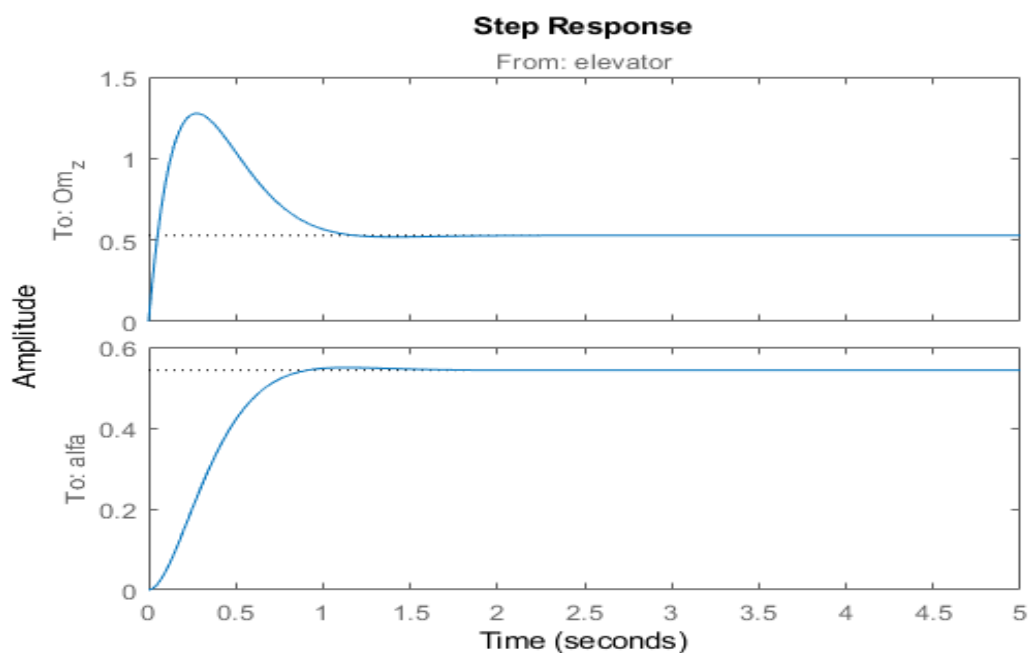


Рисунок 9.3- Кривые динамики короткопериодического продольного движения летательного аппарата

Контрольные вопросы

1. Какая функция осуществляет задание идентификаторов созданных LTI - систем во входных параметрах в MATLAB?
2. Какая функция используется для создания модели динамической системы

- в пространстве состояния в MATLAB?
3. Запишите дифференциальные уравнения в матричной форме.
 4. Как выполняется настройка графиков в MATLAB?

ЧАСТЬ III. ПАКЕТ РАСШИРЕНИЯ SIMULINK

ГЛАВА 10. ОБЩИЕ СВЕДЕНИЯ О ПРОГРАММНОМ КОМПЛЕКСЕ SIMULINK

§10.1. Основные возможности программного комплекса

Программа SIMULINK является одним из приложений к пакету MATLAB. Вместе с тем, пакет расширения SIMULINK можно рассматривать как самостоятельный продукт фирмы MathWorks. Однако, он работает только при наличии ядра MATLAB, используя входящие в его состав функции.

Пакет SIMULINK может использоваться для расчетов, функционального и имитационного моделирования различных систем связи и управления, цифровых устройств, электротехнических и энергетических установок и т.д. при автоматизации научных исследований, проектирования и управления. В основу пакета SIMULINK заложен принцип визуального программирования, позволяет создавать модели из стандартных блоков, находящихся в различных библиотеках, а также выполнять различные расчеты.

При моделировании в MATLAB позволяет осуществлять выбор метода решения задачи, изменения времени моделирования и способа задания временного шага: фиксированного или переменного. Наличие возможности вывода результатов моделирования в числовом, графическом или табличном виде обеспечивает мониторинг процессов, происходящих в системе.

Библиотека SIMULINK включает полный набор блоков, необходимых для создания функциональных моделей различных систем. В SIMULINK также имеется возможность редактирования библиотечных блоков и создания собственных. Пакет SIMULINK позволяет объединять блоки в единую систему и изучать ее поведение во времени. В основу разработки моделей средствами SIMULINK (S-модели) заложена технология drag-and-drop («перетаски

и оставь»). Для построения S-модели используются блоки или модули, хранящиеся в библиотеке SIMULINK.

Библиотека SIMULINK позволяет пользователю использовать все основные возможности пакета MATLAB. Помимо этого, библиотеку SIMULINK можно рассматривать как достаточно самостоятельную компоненту, поскольку при работе с ней не требуется наличия навыков использования других инструментов, входящих в состав пакета MATLAB.

Между блоками, входящими в состав создаваемой модели, в зависимости от типа блоков и логики работы модели могут быть установлены связи, как по информации, так и по управлению. Данные, которыми обмениваются блоки, могут иметь следующий: скалярные величины, вектора или матрицы произвольной размерности.

При построении S-моделей может использоваться иерархическая структура с практически не ограниченным числом уровней иерархии. Как отмечалось выше, пользователь может задавать способ изменения модельного времени, а также время или условия окончания моделирования.


В ходе моделирования имеется возможность наблюдать за процессами, происходящими в системе. Для этой цели используются входящие в состав библиотеки SIMULINK специальные «смотровые окна». Результаты моделирования, представляющие интерес для пользователя, могут быть представлены в числовой и графической форме.

Принципы структурного и модульного программирования, лежащие в основе MATLAB, позволяют реализовывать алгоритмы в виде набора унифицированных программных модулей. Такой подход обеспечивает удобочитаемость разрабатываемых программ, облегчает их отладку, уменьшает общий объем программного обеспечения.

§10.2. Создание нового окна модели

Модель исследуемой системы представляется в виде структурной модели и сохраняется в файле с расширением **.slx**. Для создания окна, в котором

будет осуществляться сборка новой S-модели, можно воспользоваться одним из трех способов:

1. С помощью команды **Home** \Rightarrow **New** \Rightarrow **Simulink Model**.
2. В командной строке окна **Command Window** напечатать **Simulink** нажать клавишу **Enter** на клавиатуре.
3. Нажать кнопку **Simulink**  на панели инструментов командного окна MATLAB.
4. В открывшемся окне (см. рис. 10.1) выбрать опцию **Blank Model**.

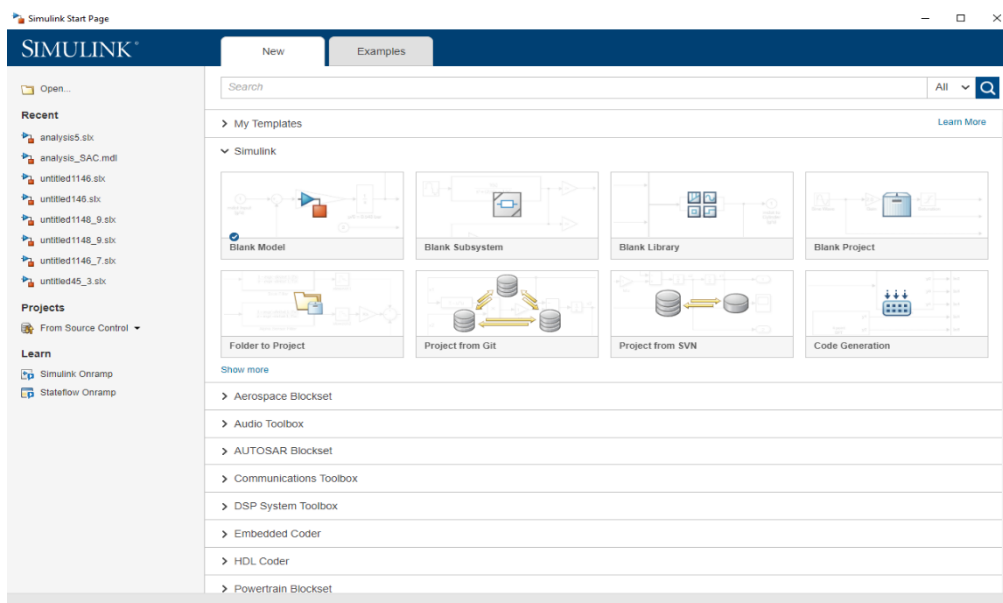


Рисунок 10.1- Начальное окно для открытия окна модели

Окно пустой модели (см. рис. 10.2.) имеет: 1- название окна (по умолчанию вновь созданному окну присваивается имя untitled с соответствующим номером); 2- меню команд; 3, 4- панели инструментов, расположенные горизонтально и сбоку; 5- окно создаваемой схемы модели; 6- строку состояния модели.

Командное меню. Командное меню (рис. 10.2) окна модели SIMULINK играет важную роль, поскольку дает пользователю более полный набор средств по управлению процессом моделирования.

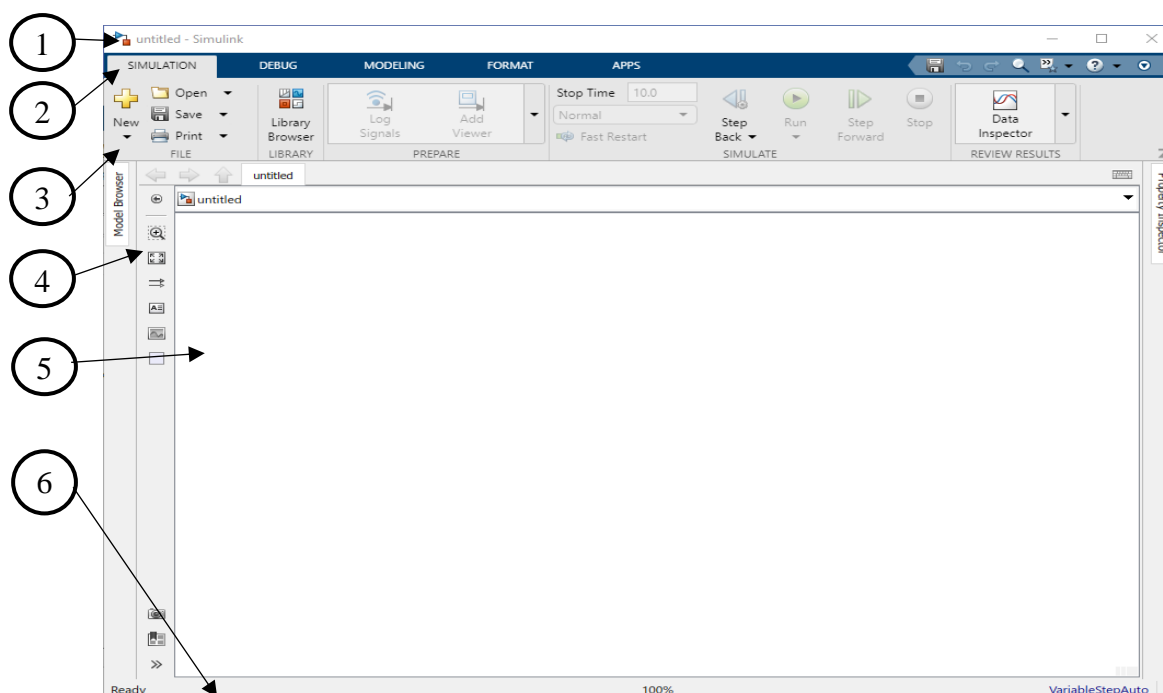
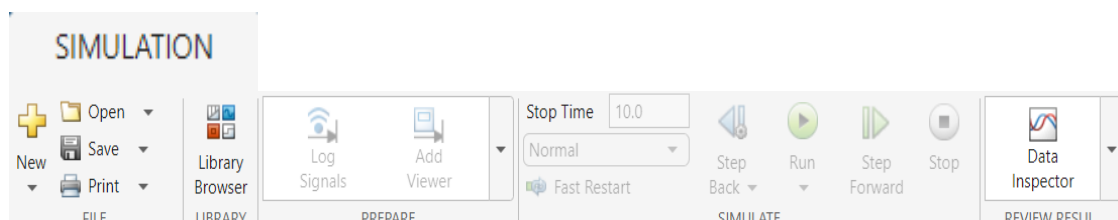
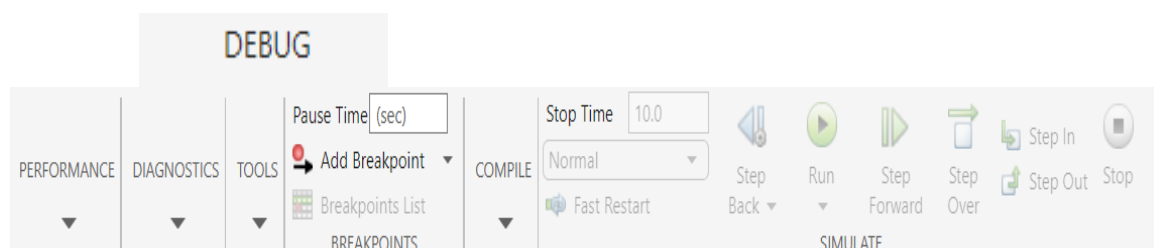


Рисунок 10.2- Пустое окно модели (окно для создания новой модели)

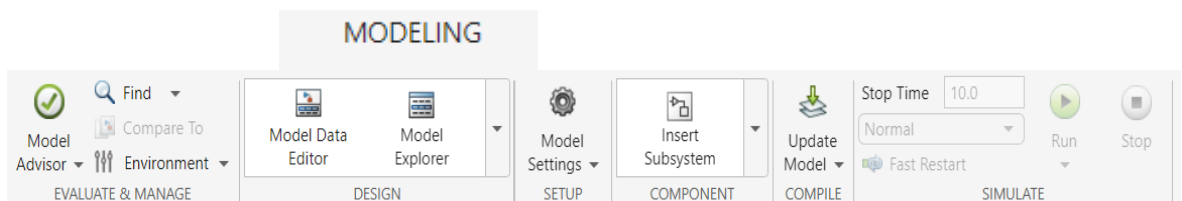
Меню содержит пять пунктов: SIMULATION, DEBUG, MODELING, FORMAT, APPS. Каждый пункт меню содержит вложенные подпункты, приведенные на рис. 10.3 а-д.



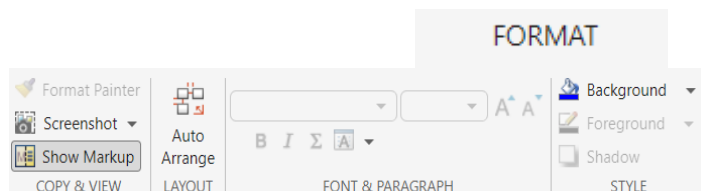
a



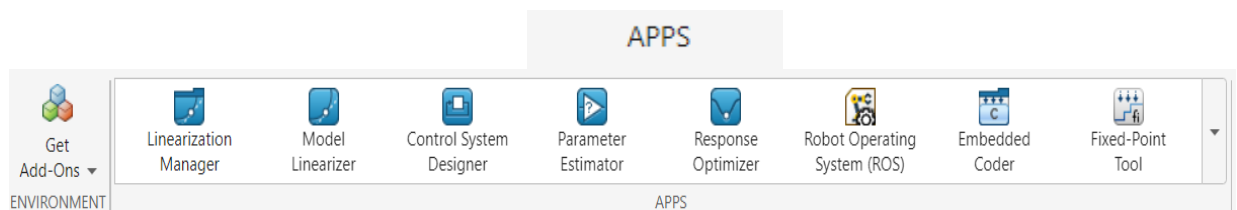
б



6



2



0

Рисунок 10.3- Вложенные подпункты меню команд


§10.3. Боковая панель инструментов

Боковая панель инструментов содержит 10 кнопок (рис. 10.4).

	Hide/Show Explorer Bar - скрыть / показать панель проводника
	Zoom - увеличить
	Fit to View - по размеру просмотра
	Sample Time - время выборки
	Annotation - аннотации
	Image - изображение
	Area - площадь
	Viewmark this View - отметить этот просмотр
	Viewmarks - метки
	Hide/Show Model Browser - скрыть / показать браузер модели

Рисунок 10.4- Боковая панель инструментов окна мо-

§10.4. Описание библиотеки SIMULINK

Библиотека SIMULINK содержит набор различных функциональных блоков. Доступ к блокам производится нажатием кнопки  **Simulink Library**, расположенной на панели инструментов MATLAB. При этом открывается окно браузера библиотеки, показанное на рис. 10.5.

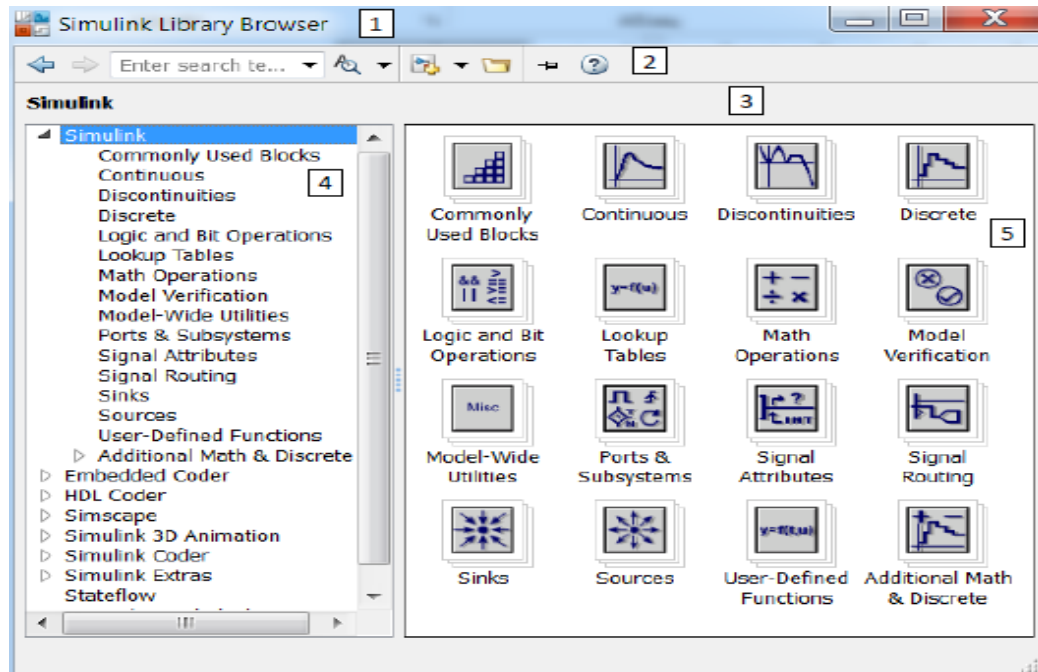


Рисунок 10.5- Окно браузера библиотеки SIMULINK

Окно браузера библиотеки SIMULINK включает следующие элементы:

- 1) **Заголовок**, содержащий название окна - **Simulink Library Browser**.
- 2) **Панель инструментов**, содержащую ярлыки наиболее часто используемых команд.
- 3) **Окно комментария**, используемое для вывода поясняющей информации о выбранном блоке.
- 4) **Список разделов библиотеки**, структурированных в виде дерева.
- 5) **Окно содержимого раздела библиотеки**.

Основная библиотека SIMULINK содержит большое количество самых разнообразных блоков, полный перечень которых имеется в **Help**. Достаточно дважды щелкнуть левой кнопкой мыши по любому блоку и в открывшемся окне параметров блока нажать кнопку **Help**.

Наиболее часто применяемые при компьютерном моделировании систем управления блоки приведены на рис. 10.6 - 10.12.

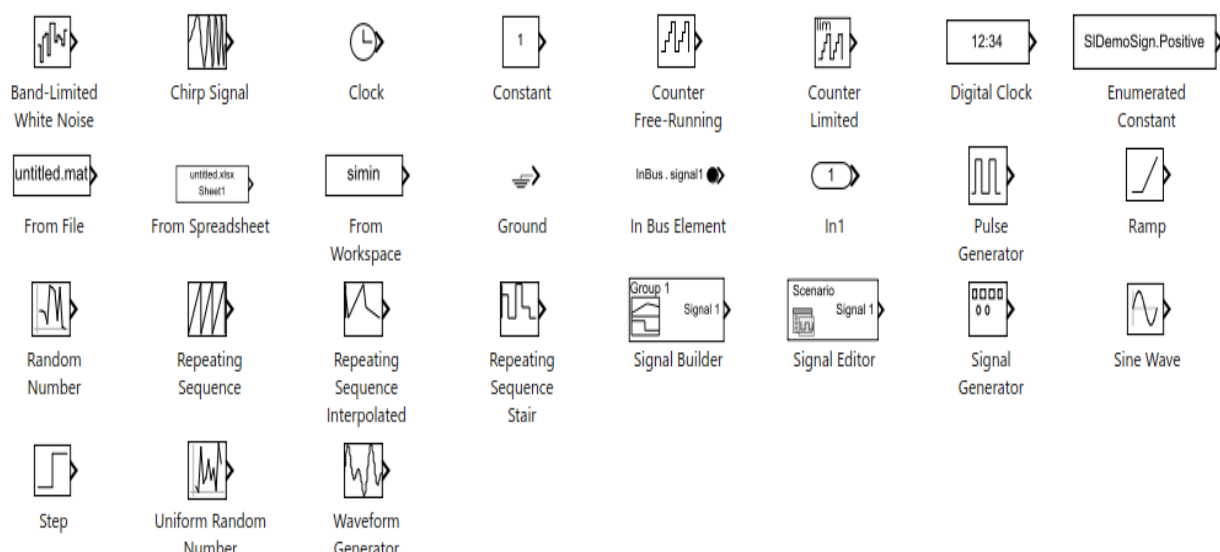


Рисунок 10.6- Раздел **Sources (источники сигналов)** библиотеки SIMULINK

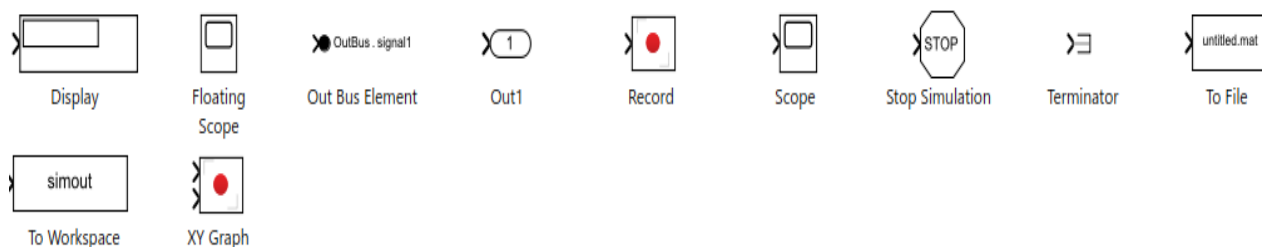


Рисунок 10.7- Раздел **Sinks (приемники сигналов)** библиотеки SIMULINK

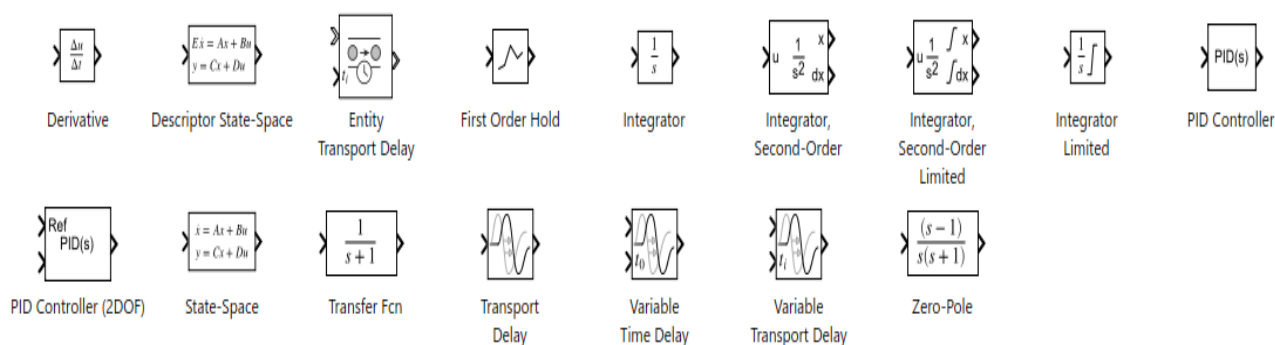


Рисунок 10.8- Раздел **Continuous (блоки непрерывных моделей)** библиотеки SIMULINK

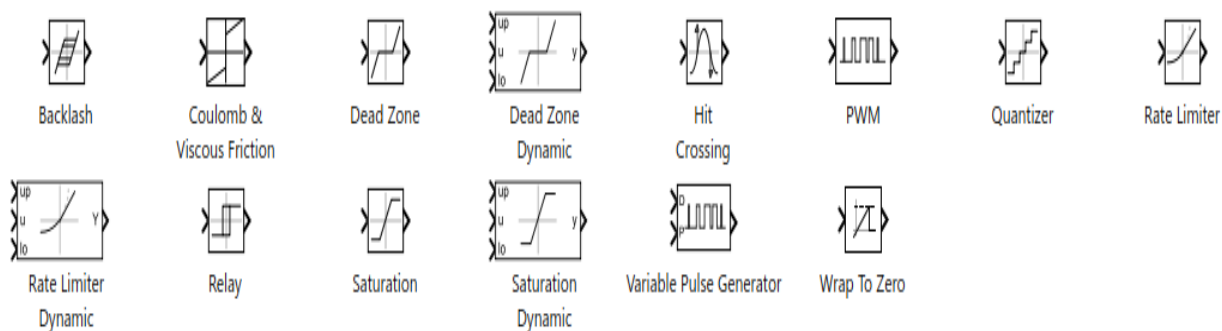


Рисунок 10.9- Раздел **Discontinuities (нелинейные блоки)** библиотеки SIMULINK

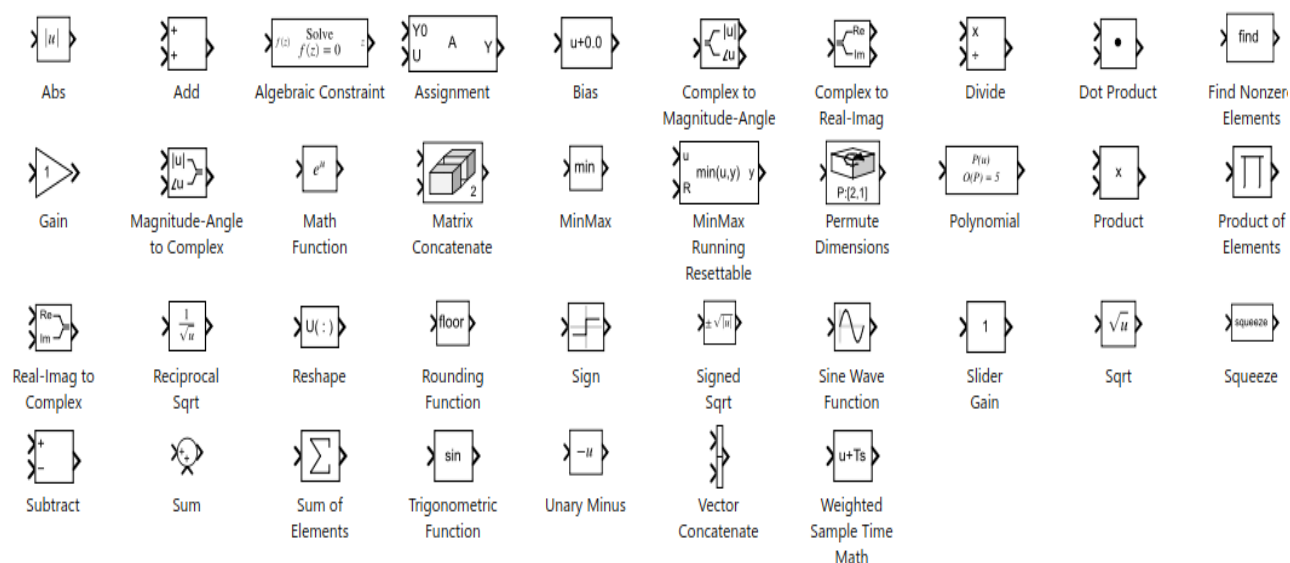


Рисунок 10.10- Раздел **Math Operations (блоки математических операций)** библиотеки SIMULINK

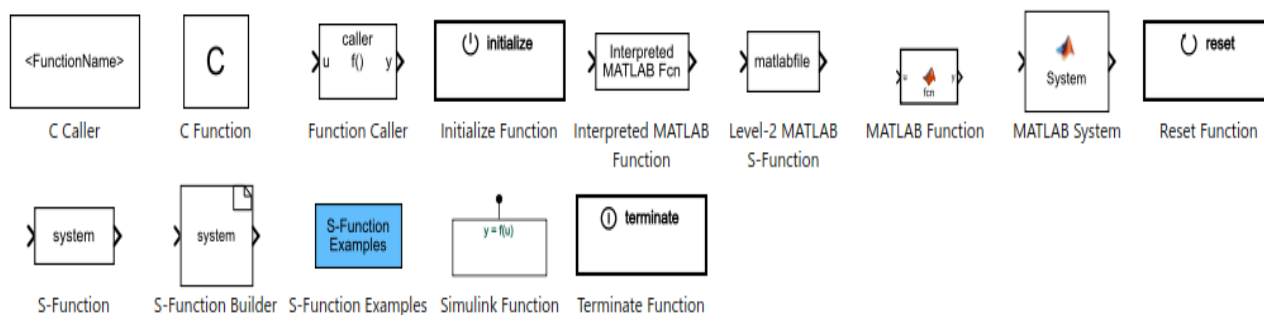


Рисунок 10.11- Раздел **User-Defined-Functions (функции, задаваемые пользователем)** библиотеки SIMULINK

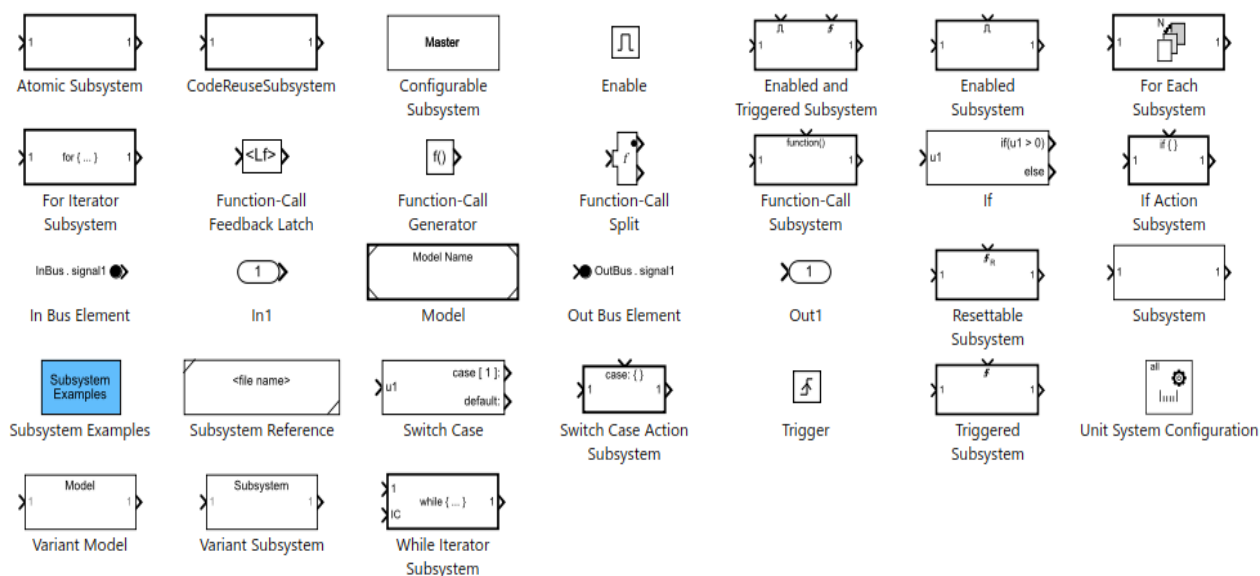


Рисунок 10.12- Раздел **Ports&Subsystems (порты и подсистемы)** библиотеки **SIMULINK**

Контрольные вопросы

1. Какие задачи могут быть решены в SIMULINK?
2. Какой принцип программирования используется в SIMULINK?
3. Какие возможности при моделировании доступны в SIMULINK?
4. Какая технология используется при разработке моделей в SIMULINK?
5. Как создается новое окно модели в SIMULINK?
6. Какое расширение имеют файлы структурной модели в MATLAB?
7. Как можно вызвать браузер библиотеки SIMULINK?
8. Какие пункты содержит командное меню SIMULINK?
9. Какие встроенные пункты содержит команда меню SIMULINK?
10. Какие функции выполняет команда меню APPS?
11. Какие кнопки и окна содержит боковая панель инструментов окна модели в SIMULINK?
12. Какие элементы включает окно браузера библиотеки в SIMULINK?

ГЛАВА 11. СОЗДАНИЕ И АНАЛИЗ МОДЕЛЕЙ САУ В ПРОГРАММЕ SIMULINK

§11.1. Создание, настройка, сохранение и вывод на печать модели

Правила размещения блоков в окне модели

Процедура размещения блоков в окне модели организуется следующим образом. В разделе библиотеки SIMULINK выбирается необходимый блок одиночным нажатием на него левой кнопкой мыши. Затем, удерживая левую кнопку мыши, осуществляется перетаскивание блока в окно модели, после чего кнопка отпускаем. Если все действия выполнены правильно, пиктограмма выбранного блока должна появиться в окне модели. Размещение нужного блока в окне модели можно также выполнить нажатием правой кнопки мыши на блоке и выбором в контекстном меню команды **Add to model** «имя модели» (Ctrl+I).

Выделение блоков в окне модели

Выделение блока производится наведением на него стрелки мыши и нажатием левой кнопки. Признаком того, что блок выделен, является появление маленьких темных прямоугольников по углам рамки блока.

Выделение группы блоков осуществляется следующим образом. Курсор мыши устанавливается рядом с выделяемыми блоками и нажимается левая кнопка мыши. При перемещении мыши появляется расширяющаяся прямоугольная рамка из тонкой линии. Выделенными оказывается каждый блок, попавший в рамку. Таким образом, выделенными окажутся все попавшие в рамку блоки. Также группу блоков можно выделить одновременным нажатием левой клавиши мыши и клавиши **Shift** на требуемых блоках.

Для выделения всех блоков модели также можно воспользоваться командой **Edit > Select All** (Ctrl+A).

Выделенную группу блоков или блок можно перетаскивать мышью, удерживая ее левую кнопку. Отпустив левую кнопку мыши, можно зафиксировать новое место группы блоков или блока.

Сохранение модели

Созданная модель может быть сохранена для последующего исследования, применения или модернизации. Для этого необходимо выполнить команду **Save** (Ctrl+S) или **Save As** меню **File** окна редактора моделей. Модель будет записана в виде файла с расширением .slx.

Установка параметров блоков

Установка параметров блоков осуществляется путем нажатия правой кнопки мыши на изображении блока и выбором в открывшемся контекстном меню команды **Block parameters**. Окно модификации параметров блока можно также открыть, дважды щелкнуть левой кнопкой мыши на изображении блока.

Установка параметров моделирования

Перед запуском модели необходимо проверить, а при необходимости, изменить установки параметров моделирования. Окно установки параметров моделирования (рис. 11.1) открывается командой **Model Configuration Parameters** (Ctrl+E) на горизонтальной панели инструментов, либо в меню **Simulation** окна SIMULINK.

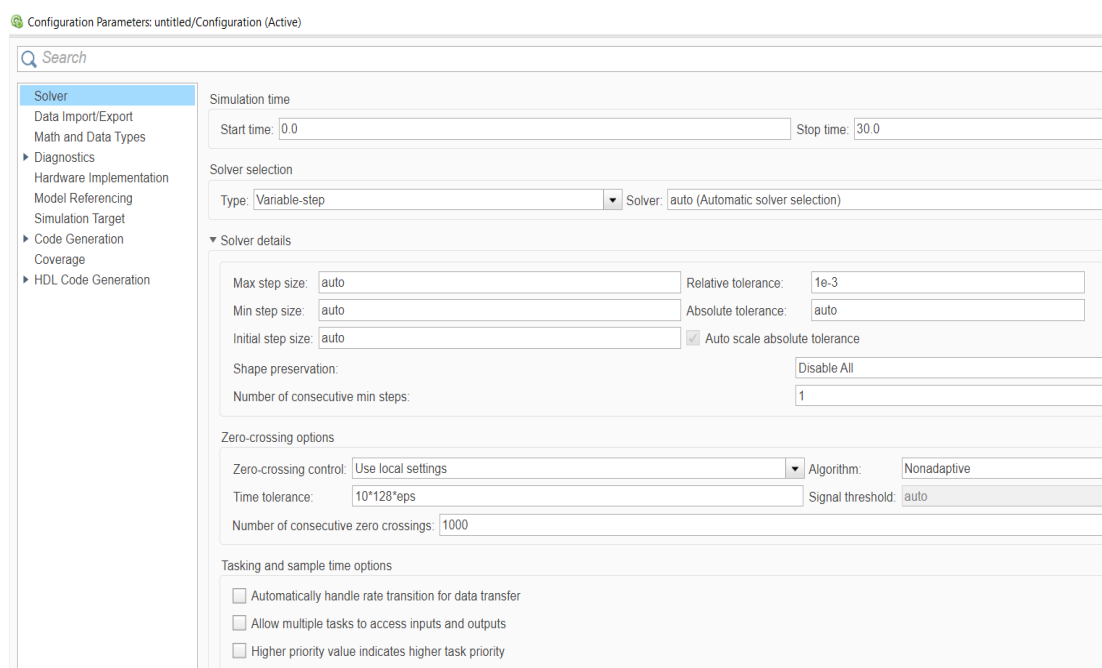


Рисунок 11.1- Окно задания параметров моделирования

Окно параметров модели имеет ряд вкладок, содержащих довольно большое число параметров.

Исходя из задач данного учебника ограничимся изучением только одной вкладки **Solver** (решатель), которая открывается по умолчанию. Вкладка **Solver** позволяет установить параметры решающего устройства системы моделирования SIMULINK. Одним из важнейших параметров решателя является время моделирования - **Simulation time**, которое задается начальным временем **Start time** (обычно 0) и конечным временем **Stop time** (по умолчанию 10). При равенстве **Stop time** бесконечности (inf) моделирование происходит бесконечно долго. Прерывание осуществляется с помощью кнопки окна модели или команды **Stop** из командного меню. Рекомендуется задавать конечные значения **Stop time**, поскольку в этом случае можно легко получить различные осциллограммы работы устройства.

Опция решателя **Solver selection** содержит:

- **Type** - тип решателя и метод моделирования.

Тип решателя может быть: **Variable step** - с переменным шагом по времени и **Fixed step** - с постоянным шагом по времени (для дискретных систем).

Следующая группа параметров изменяется от выбора типа решателя.

1). Для переменного шага моделирования имеется:

Справа от типа решателя выбирается метод моделирования, возможен выбор следующих методов: **discrete** - дискретный, **ode45** - метод Дорманда-Принса, **ode23** - метод Богацкого-Шампина, **ode113** - метод Адамса, **ode15s** - stiff/NDF, **ode23s** - stiff/Mod Rosenbrock, **ode23t** - stiff/Trapezoidal, **ode23tb** - stiff/TR-BDF2.

- **Max step size** - максимальное значение шага моделирования;
- **Min step size** - минимальное значение шага моделирования;
- **Initial step size** - начальный шаг моделирования.

По умолчанию для трех вышеописанных параметров устанавливается значение **Auto** (выбрать автоматически). Однако, это значение можно изменять и устанавливать необходимое для данной задачи.

Number of consecutive min steps - количество последовательных шагов min (по умолчанию 1).

Relative tolerance - относительная погрешность моделирования.

Absolute tolerance - абсолютная погрешность моделирования.

Shape preservation - сохранение формы (**Disable All** - все отключить, **Enable All** - все включить).

Следующая группа опций **Tasking and sample time options** - опции задачи и времени выборки содержит:

- **Tasking mode for periodic sample times** - режим задачи для периодических выборок - **Auto** (выбрать автоматически);
- **Automatically handle rate transition for data transfer** - автоматически обрабатывать изменение скорости передачи данных (выбор режима осуществляется простановкой галочки в поле пустого квадрата);
- **Higher priority value indicates higher task priority** - более высокое значение приоритета указывает на более высокий приоритет задачи.

Опции **Zero-crossing options** - пересечение нулем содержат:

- **Zero-crossing control** - контроль пересечения нулем может иметь значения: **Use local settings** - использовать локальные настройки, **Enable All** - все включить, **Disable All** - все отключить;
- **Algorithm** - алгоритм содержит: **Nonadaptive** - неадаптивный, **Adaptive** - адаптивный;
- **Time tolerance** - допуск времени, который показывает насколько близко должны происходить события перехода через ноль, чтобы их можно было считать последовательными (по умолчанию $10 \cdot 128 \cdot \text{eps}$);
- **Signal threshold** - порог сигнала - **Auto** (выбрать автоматически);
- **Number of consecutive zero crossings** - количество последовательных переходов через ноль (по умолчанию 1000).

2). Для постоянного шага моделирования имеется:

Fixed step size (fundamental sample time) - значение фиксированного шага моделирования (при значении **auto** устанавливается шаг, заданный

источником сигнала, и если источников несколько - устанавливается наименьший шаг моделирования).

Solver - решатель включает следующий набор методов: **discrete** (no continuous states) - дискретный, **ode8** - метод Дорманда-Принса, **ode5** - метод Дорманда-Принса, **ode4** - метод Рунге-Кутты, **ode3** - метод Богацкого-Шампины, **ode2** - метод Хойна, **ode1** - метод Эйлера, **ode14x** - метод экстраполяции, **ode1be**- (Бэкворд- Эйлера).

Опции **Tasking and sample time options** - опции задачи и времени выборки содержат:

- **Periodic sample time constraint** - ограничение по времени периодической выборки (**Unconstrained** - без ограничений, **Ensure sample time independent** - обеспечение независимости от времени выборки, **Specified** – определенный);
- **Tasking mode for periodic sample times** - режим задачи для периодических выборок (**Auto** - выбрать автоматически, **SingleTasking** - однозадачный режим, **MultiTasking** - многозадачный режим);
- **Automatically handle rate transition for data transfer** – автоматически обрабатывать изменение скорости передачи данных;
- **Higher priority value indicates higher task priority** - более высокое значение приоритета указывает на более высокий приоритет задачи.

Правильная установка значений всех описанных выше параметров важна при моделировании сложных систем. При моделировании линейных САУ непрерывного действия важно установить только конечное время моделирования. Остальные параметры можно оставить установленными по умолчанию автоматически. То же самое относится и к моделированию дискретных систем, необходимо лишь выбрать тип решателя - с постоянным шагом моделирования и указать шаг (при необходимости).

Создание надписей и текстовых комментариев

Хорошая программа должна содержать необходимое количество комментариев, которые позволяют без труда отслеживать логику работы

программы. То же самое относится и к моделям, созданным в пакете SIMULINK. Модели, которые не содержат текстовые комментарии, обладают малой наглядностью и трудно воспринимаются пользователями. Так же нежелательно перенасыщение моделей комментариями.

При создании текстовой надписи в поле модели необходимо указать мышью место надписи, дважды щелкнув левой кнопкой мыши. При этом откроется блок надписи с курсором ввода, куда и вводится текст комментария. После ввода надписи нажимается клавиша **Enter**.

Для изменения содержания надписи блоков моделей достаточно установить мышь в область надписи и щелкнуть левой кнопкой мыши. В надписи появится курсор ввода, после чего надпись можно редактировать. Убрать надпись можно двумя способами: 1) нужно выделить надпись и выполнить команду **Edit > Delete**; 2) установить мышь в область надписи, щелкнуть правой кнопкой мыши и в открывшемся окне выбрать опцию **Delete**. Наиболее просто выделенную надпись можно убрать нажатием клавиши **Delete** на клавиатуре.

Поскольку система MATLAB является англоязычной, при большом числе комментариев на русском языке возможны сбои в работе системы.

Соединение блоков между собой

Инструментом для соединения блоков между собой в MATLAB является мышь. Каждый блок моделей имеет входы и (или) выходы. Как правило, выход предыдущего блока подключается ко входу следующего блока. Если курсор мыши устанавливается на выходе блока, от которого должно исходить соединение, он превращается в большой крестик из тонких линий. Удерживая нажатой левую кнопку мыши, необходимо плавно переместить курсор к входу следующего блока. Курсор мыши на входе следующего блока приобретет вид крестика из тонких сдвоенных линий.

После появления соединительной линии к входу следующего блока следует отпустить левую кнопку мыши. Если соединение завершено правильно, в его конце появится жирная стрелка. Выделение соединения

производится щелчком левой кнопки мыши. Меню настройки и редактирования соединения открывается нажатием правой кнопки мыши.

Порядок создания петли соединительной линии в необходимую сторону сводится к следующему. Выполняется захват нужной части линии, после чего она отводится в нужную сторону перемещением мыши с нажатой левой кнопкой. После отпускания левой кнопки мыши создание петли линии заканчивается. Полезная возможность задания наклонных линий соединений в MATLAB связана с использованием нажатой клавиши **Shift**.

Примеры горизонтального и диагонального соединений двух блоков приведены на рис. 11.2.

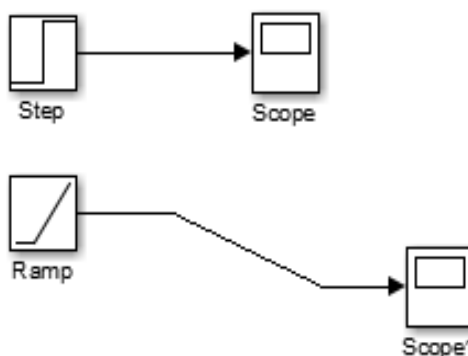


Рисунок 11.2- Соединение блоков между собой

Создание отвода линий

Часто на практике возникает необходимость сделать отводы от созданных линий. Для этого достаточно подвести курсор мыши к созданной линии и нажать правую кнопку мыши. Удерживая правую кнопку мыши, следует переместить курсор ко входу соответствующего блока. При появлении соединительной линии ко входу следующего блока необходимо отпустить кнопку мыши. После завершения соединения на его конце появится жирная стрелка.

Также, как и в предыдущем случае, построение наклонных отводов линии может быть осуществлено путем нажатия клавиши **Shift**.

Горизонтальный и наклонный отводы линии одного из примеров показаны на рис. 11.3.

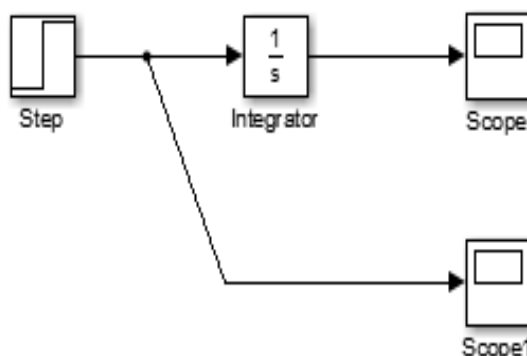


Рисунок 11.3- Соединение блоков между собой

Удаление соединений

Удаления соединительной линии производится путем ее выделения щелчком левой кнопки мыши и нажатием клавиши **Delete** на клавиатуре. Также можно навести стрелку мыши на соединительную линию и, нажав правую кнопку мыши, в открывшемся окне выполнить команду **Cut** (Ctrl+X) или **Delete** (Del).

Изменение размеров блоков

SIMULINK включает различные возможности редактирования блок-схем. Блоки в окне модели можно перемещать мышью, изменять их размеры и угол поворота. Для изменения размера необходимо выделить блок, затем курсор мыши надо установить на квадратики по углам блока. После того, как курсор мыши превратится в двунаправленную диагональную стрелку, можно растягивать блоки по диагонали, увеличивая или уменьшая их размеры. Левая кнопка мыши при этом должна оставаться нажатой.

Процедура растягивания относится только к графическому изображению блока. Размеры названия блока в форме текстовой надписи не изменяются.

Перемещение блоков и вставка блоков в соединение

Блоки в окне модели можно перемещать мышью, предварительно их выделив. При этом установленные соединения блока не прерываются, они

просто сокращаются или увеличиваются в длине. Предварительно выделенные блоки также можно перемещать нажатием клавиш со стрелками на клавиатуре.

Если соединение длинное, путем не сложных манипуляций в него можно вставить новый блок, не разрушая старое соединение. Для этого лишь нужно совместить вход и выход нового блока с линиями соединения (блок должен иметь один вход и один выход, для остальных блоков данный способ не приемлем).

Печать текущей модели

Окно печати модели открывается командой **File > Print > Print** (Ctrl+P). Окно печати модели содержит все необходимые настройки для печати текущей модели: выбор типа принтера, область печати, выбор опций печати и т.д.

Вставка модели в текстовые редакторы

Вставка модели в различные редакторы (текстовые или графические) производится входом в опцию **Edit > Copy Current View to Clipboard** и выполнением одной из двух команд: **Metafile**, **Bitmap**. Модель будет скопирована в виде картинки в буфер обмена ОС Windows и путем нажатия комбинации клавиш **Shift+Insert** может быть вставлена в открытый файл любого текстового редактора.

Создание простейшей модели

Модели, создаваемые в пакете SIMULINK, должны включать три основные части:

- 1) источник сигнала;
- 2) совокупности различных блоков, образующих модель;
- 3) приемник сигнала.

Рассмотрим пример составления простейшей модели, состоящей из источника единичного ступенчатого воздействия (блок **Step**) из раздела **Sources**, блока интегрирования (блок **Integrator**) из раздела **Continuous** и двух виртуальных осциллографов (блок **Scope**) из раздела **Sinks**.

Создание модели и ее исследование выполняется в результате следующих действий:

- 1) открывается новое окно;
- 2) в окно модели помещаются блоки **Step** из раздела **Sources**, **Integrator** из раздела **Continuous** и **Scope** из раздела **Sinks**;
- 4) производится соединение блоков **Step**, **Integrator** и **Scope** между собой, как показано на рис. 11.4;

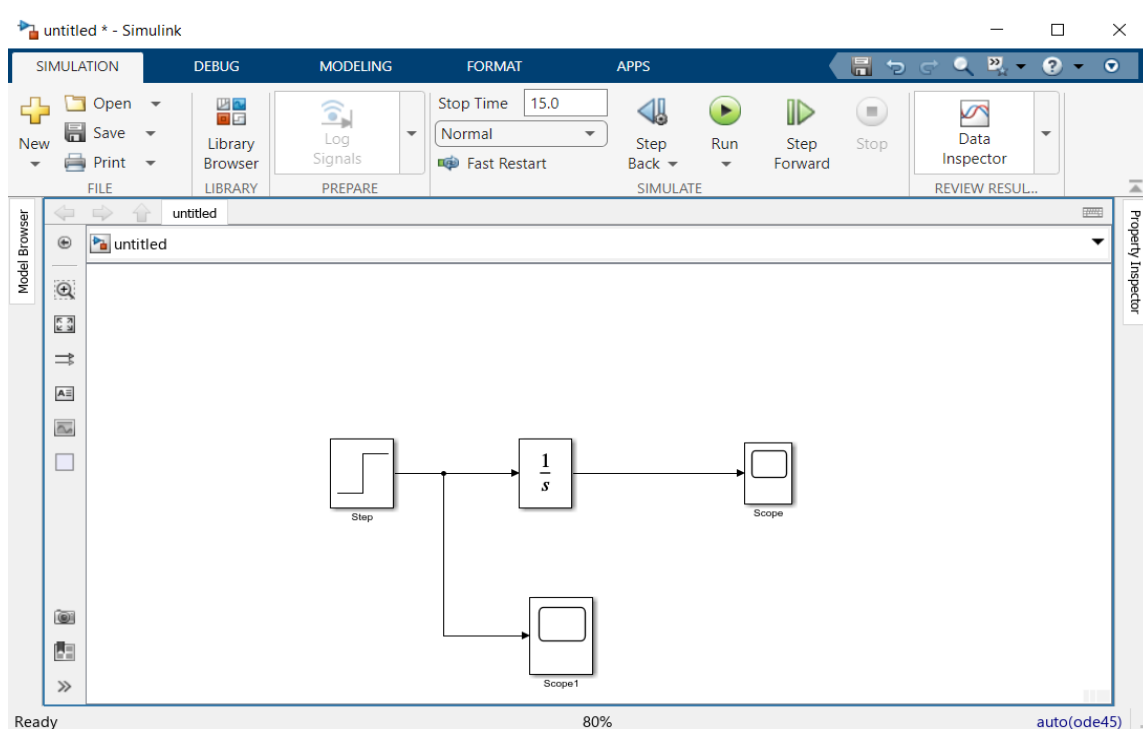


Рисунок 11.4- Простейшая модель, готовая к процессу моделирования

5) устанавливаются требуемые параметры моделирования (**Model Configuration Parameters**) **Stop time** = 15 с (остальные параметры оставляем без изменения);

6) запускается процесс моделирования нажатием кнопки  ;

7) поочередно просматриваются результаты моделирования на блоках **Scope** и **Scope1** (дважды щелкнув левой кнопкой мыши на блоках);

8) при необходимости модель сохраняется.

В окне виртуального осциллографа (блоки **Scope** и **Scope1**) имеются три области: строка команд, панель инструментов и окно осциллографа (см. рис. 11.5). При помощи команд и инструментов можно осуществлять настройку графических изображений, осуществлять запись в файл и т.д. Окно осциллографа служит для вывода результатов моделирования в графической форме.

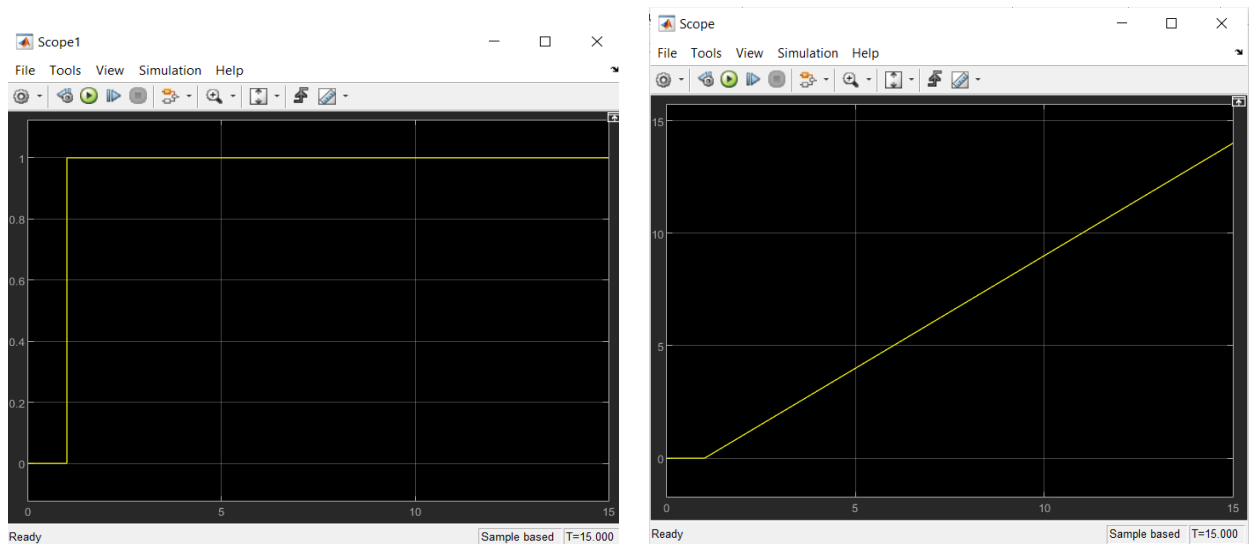


Рисунок 11.5- Окна виртуальных осциллограф **Scope** и **Scope1**

Строка команд содержит пять опций: **File**, **Tools**, **View**, **Simulation**, **Help**.

Опция **File** содержит ряд пунктов, приведенных на рис. 11.6.

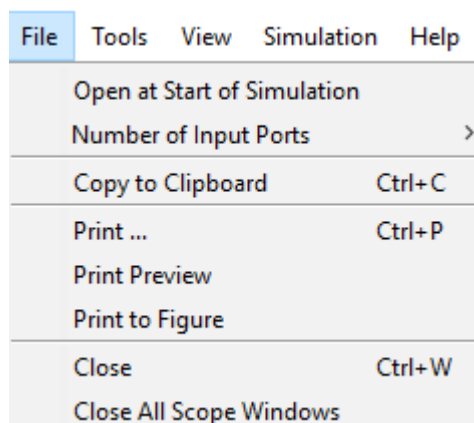


Рисунок 11.6- Пункты опции **File** меню команд

Опция **Tools** содержит ряд пунктов, приведенных на рис. 11.7.

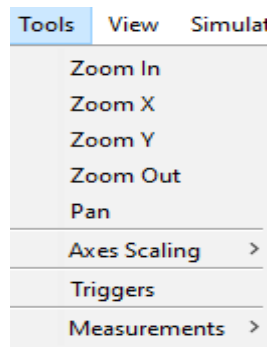


Рисунок 11.7- Пункты опции **Tools** меню команд

Опция **View** содержит ряд пунктов, приведенных на рис. 11.8.

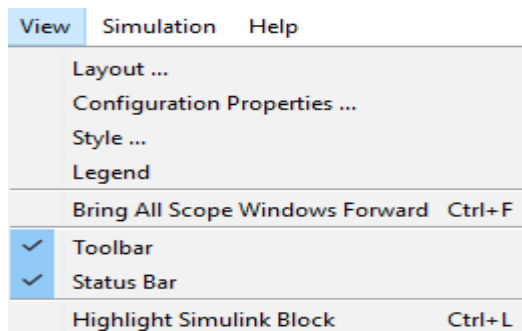


Рисунок 11.8- Пункты опции **View** меню команд

Опция **Simulation** содержит ряд пунктов, приведенных на рис. 11.9.

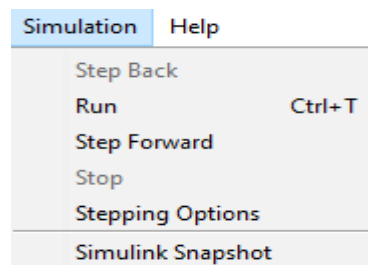


Рисунок 11.9- Пункты опции **Simulation** меню команд

Опция **Help** содержит ряд пунктов, приведенных на рис. 11.10.

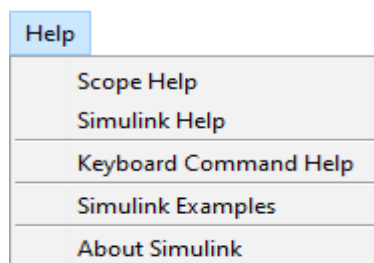


Рисунок 11.10- Пункты опции **Help** меню команд

Панель инструментов состоит из следующих кнопок (слева направо):

1. **Configuration Properties: Scope** свойства конфигурации, которая имеет четыре опции: **Configuration Properties, Stile, Layout, Show legend**;
2. **Simulation Stepping Options: untitled** - настройка пошагового моделирования;
3. **Run** - запуск программы;
4. **Step Forward** - шаг вперед;
5. **Stop** - останов программы;
6. **Highlight Simulink block** - выделение блоков Simulink, которая имеет две опции: **Highlight Simulink block, Snapshot (Freeze display)**;
7. **Zoom In** - увеличение масштаба, которая имеет четыре опции: **Zoom X, Zoom Y, Zoom Out, Pan**;
8. **Scale Y - Axis Limits** - масштабные пределы оси Y, которая имеет две опции: **Scale X - Axis Limits, Scale X&Y Axis Limits**;
9. **Triggers** - пусковая кнопка, при нажатии на которую открываются четыре опции: **Main, Source/Tupe, Levels/Timing, Delay/Holdoff** ;
10. **Cursor Measurements** - курсорные измерения, которая имеет четыре опции: **Cursor Measurements, Signal Statistics, Bilevel Measurements, Peak Finder**.

§11.2. Линеаризация и анализ моделей инструментом Linear analysis

Линеаризация моделей SIMULINK

Процедура линеаризации направлена на создание линейно аппроксимированной нелинейной системы, действующей в небольшой окрестности точки заданного режима работы. Линеаризация облегчает проектирование систем управления классическими методами: диаграммы Боде (ЛАЧХ), методы корневого годографа. Линеаризация необходима для анализа поведения системы, определения показателей качества управления системы, таких как, устойчивость системы, подавление возмущений и т.д. Линеаризация нелинейных моделей SIMULINK обеспечивает получение линейной модели в

пространстве состояний, форме передаточной функции или модели нулей и полюсов.

Инструмент **Linear analysis** позволяет в интерактивном режиме:

- осуществлять линеаризацию модели в разных рабочих точках и производить сравнительный анализ переходных процессов системы в окрестности этих точек;
- получать рабочие точки путем обрезки или моделирования моделей;
- выполнять точную линеаризацию нелинейных моделей;
- производить оценку частотных характеристик, запаса устойчивости системы линейных и нелинейных моделей;
- выполнять синтез линейных систем управления с пониженной чувствительностью к изменениям параметров и погрешностям моделирования;
- выполнять пакетную линеаризацию моделей при вариации параметров;
- создавать код MATLAB для выполнения задач линеаризации;
- создавать код MATLAB для вычисления рабочих точек.

Получение переходных и частотных характеристик линейных и нелинейных моделей в пакете SIMULINK с помощью **Linear analysis** сводится к выполнению описанных ниже действий.

Установка линейных аналитических точек

Перед началом анализа s-модели на ее входе и выходе необходимо поставить точки входа и выхода. Точки анализа могут служить одной или несколькими целям:

Input - вводить аддитивный входной сигнал в точке анализа (например, для моделирования возмущения на входе в систему);

Output - измерять значение сигнала в точке (например, для изучения влияния помехи на выход системы);

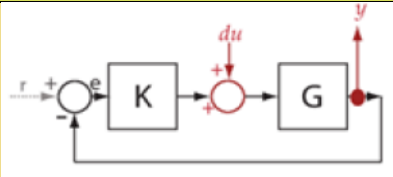
Loop Opening - интерпретировать разрыв в сигнальном потоке в точке (например, для изучения реакции разомкнутого контура на входе в систему).

Для анализа линейной модели части системы необходимо на входном и выходном сигнале этой части модели указать входную точку линеаризации и точку вывода. Применительно к анализу системы с разомкнутым контуром необходимо разбить поток сигнала, задав петлевые интервалы. Указав несколько точек ввода и вывода, можно также определить линейные модели МИМО.


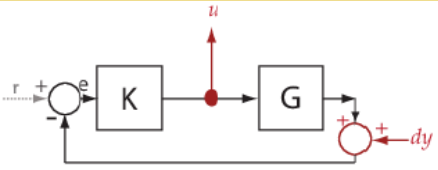

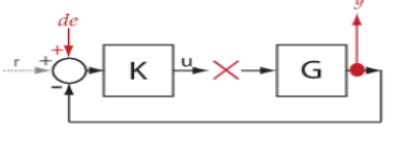

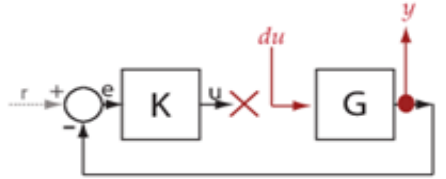

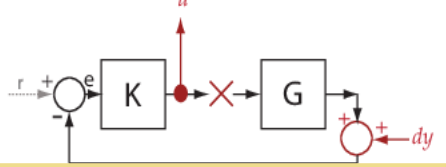
Описание точек **Linear analysis** систематизировано в таблице 11.1. Подключение аналитических точек показано на рис. 11.11.

При определении переходных характеристик в открывшемся окне следует выбрать точку входа **Open-loop Input** и точку выхода **Open-loop Output** (рис. 11.7). Выходная точка **Open-loop Output** является точкой с разомкнутым контуром, который соответствует выходному измерению с последующим открытием цепи. При линеаризации удаляются эффекты сигнала обратной связи без изменения рабочей точки модели.

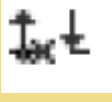
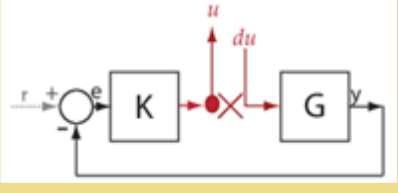

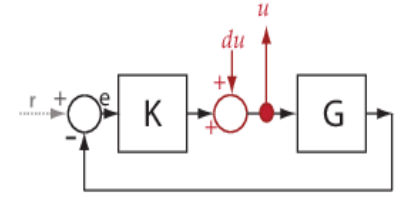
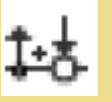
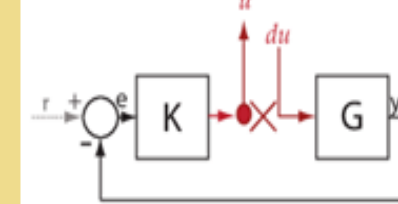
Таблица 11.1 - Типы точек линейного анализа **Control Design**

Рабочие точки линейного анализа	Описание	Схема подключения
1	2	3
Input perturbation 	Определяется аддитивный вход сигнала	 <p>Для вычисления выходного сигнала системы $G/(1+GK)$ следует указать рабочую точку входного возмущения du и выходное измерение y.</p>

Продолжение таблицы 1

1	2	3
Output measurement 	<p>Осуществляет прием измерения по сигналу. Определение передаточной функции линеаризованной системы осуществляется с использованием выходного измерения с входным возмущением или входом с разомкнутым контуром.</p>	 <p>Для вычисления $-K/(1+KG)$ в системе, следует указать выходную точку измерения u и входное возмущение dy.</p>
Loop break 	<p>Применяется для вычисления передаточной функции разомкнутой системы. Используется в случае, когда есть вложенные цепи или нужно игнорировать эффект некоторых цепей.</p>	 <p>Разрыв цепи в системе останавливает поток сигнала u. Передаточная функция от входного возмущения de к выходному измерению y в результате равна 0.</p>
Open-loop input 	<p>Прерывает цепь, за которым следует входное возмущение, для чего используется вход с разомкнутым контуром с выходным измерением или выходом с разомкнутым контуром.</p>	 <p>Необходимо добавить в системе входной сигнал du разомкнутого контура перед G и выходное измерение y после G. Вход с разомкнутым контуром прерывает поток сигнала u и добавляет входное возмущение du.</p>
Open-loop output 	<p>Осуществляет определение выходного измерения, за которым следует разрыв цепи, для чего используется выходной сигнал с разомкнутым контуром с входным возмущением или входом с открытым контуром.</p>	 <p>Чтобы определить влияние $-K$, следует добавить выходной сигнал в разомкнутом контуре после K и входное возмущение dy после G. Выход с разомкнутым контуром прерывает поток сигнала и добавляет выходное значение u.</p>

Продолжение таблицы 1

1	2	3
<p>Loop transfer function</p> 	<p>Задаёт выходное измерение перед разрывом контура, за которым следует входное возмущение. Чтобы вычислить передаточную функцию с открытым контуром вокруг цикла используется точка анализа переноса цикла.</p>	 <p>Для вычисления $-KG$ в системе следует указать точку анализа переноса цепи. Программа добавляет выходное измерение и разрывает поток сигнала с добавлением входного возмущения du.</p>
<p>Sensitivity function</p> 	<p>Определяется входное возмущение, за которым следует измерение выхода. Производится измерение чувствительности сигнала к дополнительной помехе, которая является замкнутой мерой. Обратная связь уменьшает чувствительность в полосе частот, в которой коэффициент разомкнутого контура больше 1.</p>	 <p>В систему следует включить точку анализа функции чувствительности. Программа добавляет входное возмущение du, за которым следует выходное измерение u. Передаточная функция замкнутого контура из du в u равна $1/(1 + GK)$.</p>
<p>Complementary sensitivity function</p> 	<p>Определяется выходное измерение, за которым следует входное возмущение. Дополнительная функция чувствительности в точке является передаточной функцией от аддитивного возмущения в точке до измерения в одной и той же точке. Помехи добавляются после измерения. Применяется для вычисления передаточной функции замкнутого контура.</p>	 <p>Для вычисления $-KG$ в системе следует задать точку анализа переноса цепи. Программа добавляет выходное измерение, разрывает поток сигнала и добавляет входное возмущение du.</p>

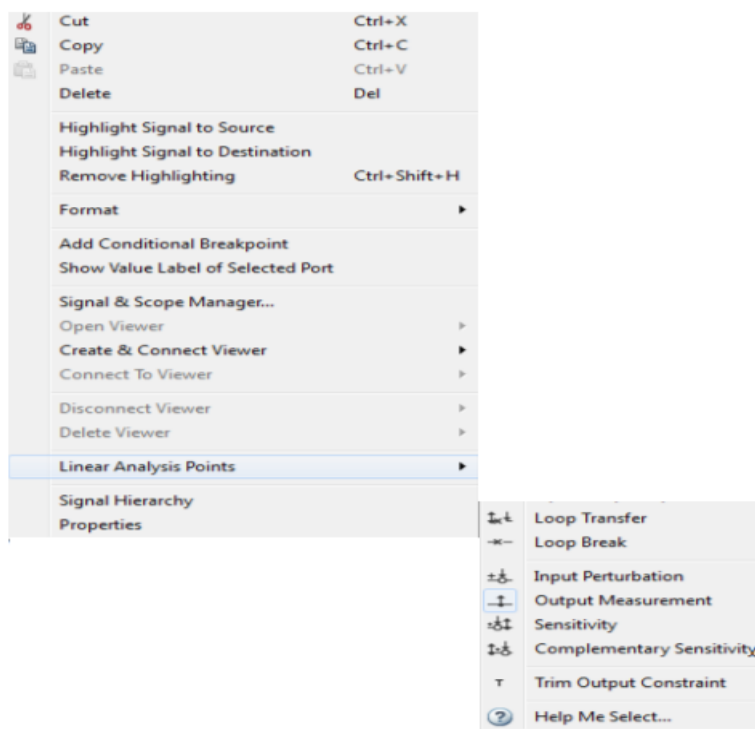


Рисунок 11.11- Подключение аналитических точек **Input Point** и **Output Point**

Определение типа характеристики для линейного анализа

Используя меню **Analysis**, выбирается пункт **Control Design**, в котором запускается функция **Linear Analysis Tool**.

После выполнения этих действий откроется окно приложения **Linear Analysis Tool** (рис. 11.12). В открывшемся окне в меню **Linearize** выбирается одна из команд, соответствующая определенному типу характеристик:

Step - переходный процесс при единичном ступенчатом воздействии (график **Step Response**);

Bode - амплитудная (ЛАЧХ) и фазовая (ЛФЧХ) логарифмические частотные характеристики при гармоническом воздействии (график **Bode Diagram**);

Impulse - импульсная переходная характеристика при единичном импульсном воздействии (график **Impulse Response**);

Nyquist - амплитудно-фазовая частотная характеристика АФЧХ при гармоническом воздействии (график **Nyquist Diagram**);

Nichols - частотная характеристика Николса (график **Nichols Chart**);

Singular Valurs - сингулярные значения (график **Singular Valurs**);

Pole/Zero Map - диаграмма распределения полюсов-нулей передаточной функции модели (график **Pole-Zero Map**);

I/O Pole/Zero Map - диаграмма распределения полюсов-нулей передаточной функции модели (график **Pole-Zero Map**).

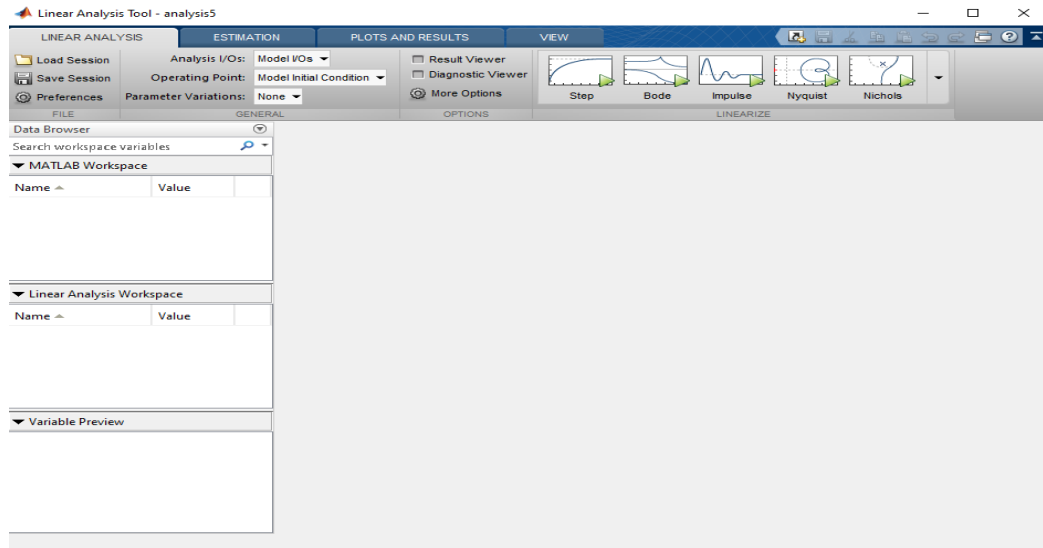


Рисунок 11.12- Окно приложения **Linear Analysis**

Настройка параметров характеристик

После появления в основном поле графика характеристики возникает возможность для редактирования и просмотра графиков в контекстном меню (рис. 11.13). Контекстное меню открывается нажатием правой кнопки мыши на поле графика. Для просмотра характеристик системы (границы устойчивости, перерегулирование или время регулирования) следует щелкнуть правой кнопкой мыши на графике и выбрать окно **Characteristics**.

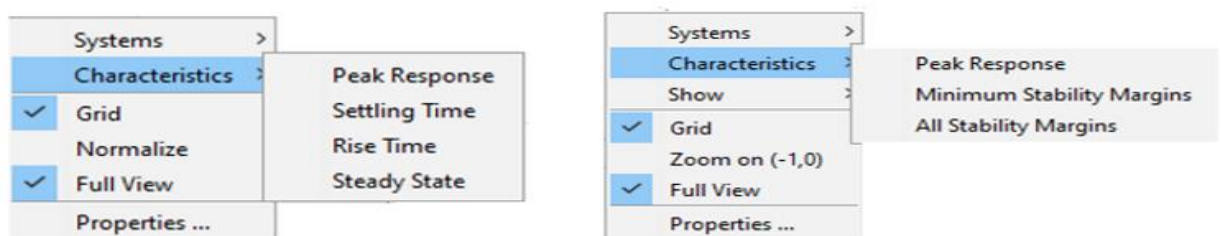


Рисунок 11.13- Настройки окна **Characteristics**

Параметры для переходных характеристик:

- **Peak Response** - максимальное значение, перерегулирование и время, на котором отмечено максимальное значение выходной величины;

- **Settling Time** - время переходного процесса;
- **Rise Time** - время нарастания выходного сигнала;
- **Steady State** - установившееся значение переходного процесса.

Параметры для частотных характеристик:

- **Peak Response** - максимальное значение амплитуды и частота, на которой зафиксировано это значение;
- **Minimum Stability Margins** - точка устойчивости (минимум), отображается **phase margin** - запас по фазе и **delay margin**- запас по запаздыванию, частота **at frequency** в соответствующей точке, вывод об устойчивости системы в замкнутом виде.
- **All Stability Margins** - точки устойчивости (все).

Характеристики на графике отображаются метками данных. При наведении на метку указателем мыши отображается подсказка, которая содержит информацию о характеристике системы.

Контрольные вопросы

1. Как осуществляется размещение блоков в окне модели SIMULINK?
2. Как осуществляется выделение блока или группы блоков в окне модели SIMULINK?
3. Как осуществляется сохранение модели в SIMULINK?
4. Как осуществляется установка параметров моделирования в SIMULINK?
5. Как осуществляется добавление надписей и текстовых комментариев в окне модели SIMULINK?
6. Как осуществляется соединение блоков между собой, создание отвода линий и удаление соединений в окне модели SIMULINK?
7. Как осуществляется изменение размеров блоков, их перемещение и вставка блоков в соединение в окне модели SIMULINK?
8. Как осуществляется печать текущей модели и вставка модели в текстовые редакторы в SIMULINK?

ЧАСТЬ IV. ОПИСАНИЯ ПРАКТИЧЕСКИХ ЗАНЯТИЙ

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ №1

Простейшие вычисления в MATLAB

Цель занятия: изучение на практике реализации средствами MATLAB вычислений математических функций.

Задание

1. Вычислить значение выражения, заданного в варианте.

Пример

$$\frac{(\sin 9,15\pi - \cos 4,35\pi)^2}{2\operatorname{ctg} 6 - \operatorname{tg} 5 \cdot \sin 3,6\pi} + \sqrt{\ln 15,6} \cdot e^{-1/5}$$

Код программы в MATLAB

```
(sin(9.15*pi)-cos(4.35*pi))^2/(2*cot(6)-tan(5)*sin(3.6*pi))+sqrt(log(15.6))*exp(-1/5)
```

ans =

1.2753

2. Найдите значение выражения, разделив на части и используя присвоение переменных.

Пример

$$\frac{\frac{\operatorname{ctg} 3,15}{\lg 6,45} - \sqrt{\frac{\lg 5,8}{\sin 3,5\pi} + \frac{\cos 2,6\pi}{\lg 1,5}}}{\frac{\operatorname{ctg} 2,15}{\lg 6,45} + \frac{\cos 2,5\pi}{\lg 1,5}};$$

Код программы в MATLAB

```
x=cot(3.15)/log10(6.45);  
y=sqrt(log10(5.8)/sin(3.5*pi)+cos(2.6*pi)/log10(1.5));  
z=cot(2.15)/log10(6.45)+cos(2.5*pi)/log10(1.5);  
l=(x-y)/z  
l =  
-1.8186e+02 + 1.9643e+00i
```

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ №2

Работа с вектор-столбцами и вектор-строками

Цель занятия: изучение на практике принципов работы с вектор-столбцами и вектор-строками в MATLAB.

Задание

1. Вычислить сумму векторов $a = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}$ и $b = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}$.

Пример

$$a = \begin{pmatrix} 1.5 \\ 4.6 \\ 105 \end{pmatrix}; b = \begin{pmatrix} 5.6 \\ 1.5 \\ 4.6 \end{pmatrix}$$

Код программы в MATLAB

```
a=[1.5;4.6;105];  
b=[5.6;1.5;4.6];  
c=a+b  
c =  
    7.1000  
    6.1000  
   109.6000
```

2. Вывести пятый элемент вектор-строки $a = (a_1 \ a_2 \ a_3 \ a_4 \ a_5 \ a_6 \ a_7 \ a_8 \ a_9)$, заменить пятый элемент вектор-строки a на заданное число N , записать в массив пятый, первый и четвертый элементы.

Пример

$$a = (1 \ 2 \ 4 \ 5 \ 6 \ 8 \ 9 \ 5 \ 6); N = 105.6.$$

Код программы в MATLAB

```
a=[1 2 4 5 6 8 9 5 6];  
a(5)  
a(5)=105.6  
b=[a(5); a(1); a(4)]  
ans =  
    6
```

```

a =
1.0000 2.0000 4.0000 5.0000 105.6000 8.0000 9.000 5.000 6.000
b =
105.6000
1.0000
5.0000

```

3. В массиве вектора-строки $a = (a_1 \ a_2 \ a_3 \ a_4 \ a_5 \ a_6 \ a_7 \ a_8 \ a_9)$ заменить нулями элементы с третьего по восьмой; создать новый массив $b1$, используя элементы массива a с первого по пятый и составить массив $b2$, содержащий элементы a , кроме пятого (используя сцепление строк).

Пример

$$a = (1 \ 2 \ 4 \ 5 \ 6 \ 8 \ 9 \ 5 \ 6)$$

Код программы в MATLAB

```

a=[1 2 4 5 6 8 9 5 6];
a(3:8)=0;
a
b1=a(1:5)
b2=[a(1:4) a(6:9)]
a =
     1     2     0     0     0     0     0     0     6
b1 =
     1     2     0     0     0
b2 =
     1     2     0     0     0     0     0     6

```

4. Перемножить элементы вектора-столбца

$$a = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \end{pmatrix}, \text{ найти минимальный и максимальный элемент вектора } a, \text{ ин-}$$

дексы минимального и максимального элементов вектора a .

Пример

$$a = \begin{pmatrix} 1.5 \\ 3.6 \\ 4.5 \\ 5.6 \\ 4.6 \\ 10 \end{pmatrix}$$

Код программы в MATLAB

```
a=[1.5; 3.6; 4.5; 5.6; 4.6; 10];
p=prod(a) %Перемножение элементов вектора-столбца или вектора-
строки
% осуществляется при помощи функция prod
Max=max(a) % Для нахождения максимума из элементов вектора
% служит функция max
Min=min(a) %Для нахождения минимума из элементов вектора
% служит функция min
[Max,k]=max(a) %Вывод порядкового номера максимального элемента
[Min,k]=min(a) %Вывод порядкового номера минимального элемента

p =
    6.2597e+03
Max =
    10
Min =
    1.5000
Max =
    10
k =
     6
Min =
    1.5000
k =
     1
```

5. Упорядочить, вектор-строку $a = (a_1 \ a_2 \ a_3 \ a_4 \ a_5 \ a_6)$:

- а) по возрастанию;
- б) по убыванию;
- в) в порядке возрастания их модулей;
- г) по возрастанию с двумя выходными аргументами.

Пример

$$a = (8.6 \ -4.5 \ 2.6 \ -1.4 \ 0.6 \ 5.8)$$

Код программы в MATLAB

```
a=[8.6 -4.5 2.6 -1.4 0.6 5.8];
R=sort(a) %Функция упорядочения вектора по возрастанию его
          %элементов sort
R1=-sort(-a) %Функция упорядочения вектора по убыванию его
            %элементов sort
R2=sort(abs(a)) %Упорядочение элементов в порядке
               %возрастания их модулей
[rs,nd]=sort(a) %Упорядочение элементов по возрастанию с
               %двумя выходными аргументами
R =
    -4.5000    -1.4000     0.6000     2.6000     5.8000     8.6000
R1 =
     8.6000     5.8000     2.6000     0.6000    -1.4000    -4.5000
R2 =
     0.6000     1.4000     2.6000     4.5000     5.8000     8.6000
rs =
    -4.5000    -1.4000     0.6000     2.6000     5.8000     8.6000
nd =
     2         4         5         3         6         1
```

б. Выполнить следующие действия с вектор-строками $a = (a_1 \ a_2 \ a_3 \ a_4)$

и $b = (b_1 \ b_2 \ b_3 \ b_4)$:

а) перемножить вектор-строки;

б) возвести во вторую степень вектор $a = (a_1 \ a_2 \ a_3 \ a_4)$;

в) все элементы вектора $a = (a_1 \ a_2 \ a_3 \ a_4)$ возвести в степень, равную соответствующим элементам вектора $b = (b_1 \ b_2 \ b_3 \ b_4)$;

г) разделить вектор $a = (a_1 \ a_2 \ a_3 \ a_4)$ на $b = (b_1 \ b_2 \ b_3 \ b_4)$;

д) к вектору- строке $e = (e_1 \ e_2 \ e_3 \ e_4)$ прибавить число N , вычесть его из вектора, умножить вектор на число L и разделить на это же число.

Пример

$$a = (10 \ 2 \ -5 \ 4); \quad b = (4 \ 5 \ -6 \ 2); \quad e = (2 \ 5 \ 9 \ 6); \quad N=3,6; \quad L=2.$$

Код программы в MATLAB

```
a=[10 2 -5 4];
b=[4 5 -6 2];
e=[2 5 9 6];
N=3.6;
L=2;
b1=a.*b %Перемножение вектор-строк
b2=a.^2 %Возведение во вторую степень
b3=a.^b %Элементы вектора a возвести в степень, равную соответ-
ствующим
%элементам вектора b
b4=a./b %Деление вектора a на вектор b
b5=e+N %К вектору-строке прибавить число 3.6
b6=e-N %От вектора-строки отнять число 3.6
b7=e*L %Вектор-строку умножить на число 2
b8=e/L %Вектор-строку разделить на число 2
b1 =
    40     10     30      8
b2 =
   100      4     25     16
b3 =
  1.0e+04 *
    1.0000    0.0032    0.0000    0.0016
b4 =
    2.5000    0.4000    0.8333    2.0000
b5 =
    5.6000    8.6000   12.6000    9.6000
b6 =
   -1.6000    1.4000    5.4000    2.4000
b7 =
     4     10     18     12
b8 =
    1.0000    2.5000    4.5000    3.0000
```

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ №3

Операции над массивами и матрицами

Цель занятия: изучение на практике принципов работы с массивами и матрицами в MATLAB.

Задание

1. Выведите таблицу значений функции $y(x)$ в заданных точках.

Пример

$$y(x) = \frac{\operatorname{tg} 3,6x + \sqrt{5 + \sin 4,6x}}{(\sin 1,5x - 5\operatorname{tg} 3,6x)^4} - e^{-x}; \quad x = 0.1 \ 0.5 \ 0.8 \ 1.2 \ 1.5 \ 1.8$$

Код программы в MATLAB

```
x=[0.2 0.5 0.8 1.2 1.5 1.8];  
y=(tan(3.6.*x)+sqrt(5+sin(4.6.*x)))/(sin(1.5.*x)-  
5*tan(3.6.*x)).^4-exp(-x)  
y =  
    -0.8070    -0.6065    -0.3797    -0.3009    -0.2226    24.8249
```

2. Заполните вектор-столбец элементами, начинающимися с нуля и до a с шагом h (используя операцию транспонирование) и вектор-строку, начинающуюся с единицы до четырех с шагом $-h2$.

Пример

$h=0.4$; $h2=0.5$; $a=0.8$

Код программы в MATLAB

```
x=[0:0.4:0.8]'  
y=[4:-0.5:1]  
x =  
     0  
    0.4000  
    0.8000  
y =  
    4.0000    3.5000    3.0000    2.5000    2.0000    1.5000  
    1.0000
```

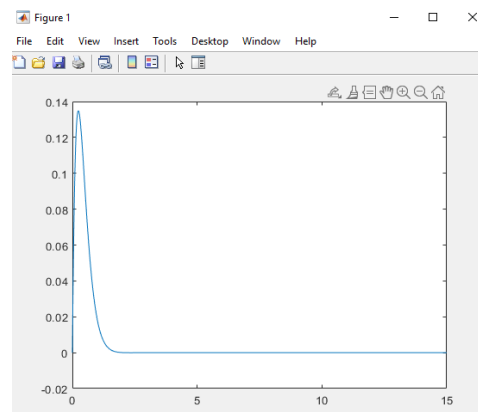
3. Построить график функции $y(x)$ на отрезке $[a, b]$ с шагом h .

Пример

$y(x) = e^{-4x} \cdot \sin 1.5x$; $x \in [0, 15]$; $h=0.01$

Код программы в MATLAB

```
x=[0:0.01:15];  
y=exp(-4.*x).*sin(1.5.*x);  
plot(x,y)
```



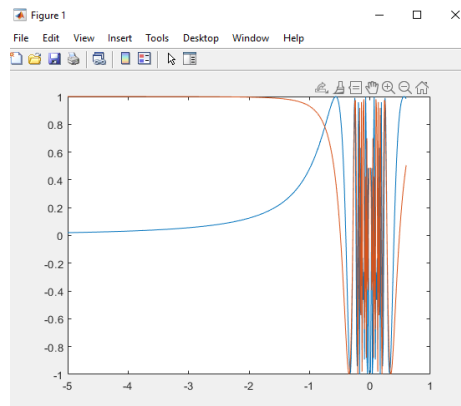
4. Постройте на отрезке $[a1, b1]$ графики функций $y(x)$ и $z(x)$.

Пример

$$y(x) = \sin \frac{1}{2x^2}; \quad z(x) = \cos \frac{1.5}{4x^2}; \quad x \in [-5, -0.6]$$

Код программы в MATLAB

```
x=[-5:0.01:0.6];  
y=sin(1./(2.*x.^2));  
z=cos(1.5./(4.*x.^2));  
plot(x,y,x,z)
```



5. Найдите скалярное произведение векторов: $a = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}$, $b = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$.

Пример

$$a = \begin{bmatrix} 1.8 \\ 4 \\ 0.6 \end{bmatrix}; \quad b = \begin{bmatrix} 5.4 \\ -4.6 \\ 1.5 \end{bmatrix}$$

Код программы в MATLAB

```
a=[1.8 4 0.6];  
b=[5.4 -4.6 1.5];  
e=dot(a,b)  
e5=sum(a.*b)  
e =  
    -7.7800  
e5 =  
    -7.7800
```

6. Найдите длину (модуль) вектора a .

Пример

$$a = \begin{bmatrix} 1.8 \\ 4 \\ 0.6 \end{bmatrix}$$

Код программы в MATLAB

```
a=[1.8;4;0.6];  
e=sqrt(sum(a.*a))  
e =  
    4.4272
```

7. Найдите векторное произведение векторов a и b .

Пример

$$a = \begin{bmatrix} 1.8 \\ 4 \\ 0.6 \end{bmatrix}; b = \begin{bmatrix} 5.4 \\ -4.6 \\ 1.5 \end{bmatrix}$$

Код программы в MATLAB

```
a=[1.8;4;0.6];  
b=[5.4;-4.6;1.5];  
e=cross(a,b)  
e =  
    8.7600  
    0.5400  
   -29.8800
```

8. Найдите объем параллелепипеда, если $a = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}$, $b = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$, $c = \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}$.

Пример

$$a = [8.6; -4.5; 2.6]; b = [-1.4; 0.6; 5.8] \quad c = [-0.6; -4.5; 5.6]$$

Код программы в MATLAB

```
a=[8.6;-4.5;2.6];  
b=[-1.4;0.6;5.8];  
c=[-0.6;-4.5;5.6];  
e=abs(sum(a.*cross(b,c)))  
e =  
    251.0520
```

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ №4

Работа с двумерными массивами и матрицами

Цель занятия: изучение на практике принципов работы с двумерными массивами и матрицами в MATLAB.

Задание

1. Найдите сумму и разность матриц A и B .

Пример

$$A = \begin{pmatrix} 6 & 5 & -7 \\ -10 & -4 & 8 \end{pmatrix}; B = \begin{pmatrix} 5 & 8 & -10 \\ 15 & 6 & 0 \end{pmatrix}$$

Код программы в MATLAB

```
A=[6 5 -7;-10 -4 8];  
B=[5 8 -10;15 6 0];  
E=A+B  
E5=A-B  
E =  
    11     13    -17  
     5      2      8  
E5 =  
     1     -3      3  
    -25    -10      8
```

2. Умножить матрицы A и Z .

Пример

$$A = \begin{pmatrix} 6 & 5 & -7 \\ -10 & -4 & 8 \end{pmatrix}; Z = \begin{pmatrix} 5 & -10 \\ 35 & 4 \\ 45 & 0 \end{pmatrix}$$

Код программы в MATLAB

```
A=[6 5 -7;-10 -4 8];  
Z=[5 -10;35 4;45 0]';  
E=A.*Z  
E =  
    30    175   -315  
   100   -16      0
```

3. Полученную матрицу умножить на число N .

Пример

$N=5$

Код программы в MATLAB

```
E=[30 175 -315;100 -16 0];  
E15=E*5  
E15 =  
      150      875     -1575  
      500     -80         0
```

4. Найти значение выражения: $(A-B)Z^4(A+B)^T$.

Код программы в MATLAB

```
A=[6 5 -7;-10 -4 8];  
B=[5 8 -10;15 6 0];  
Z=[5 -10;35 4;45 0]';  
E=(A-B).*Z.^4*(A+B)'  
E =  
    -267649375     89414375  
    -2783280     -1255120
```

5. Вычислить заданное выражение.

Пример

$$[5 \ 6 \ -10] \begin{pmatrix} 6 & 5 & 0 \\ -10 & 5 & -2 \\ 8 & 4 & -5 \end{pmatrix} \begin{bmatrix} -5 \\ 4 \\ 1 \end{bmatrix}$$

Код программы в MATLAB

```
a=[5 6 -10];  
B=[6 5 0;-10 5 -2;8 4 -5];  
z=[-5 4 1]';  
E=a*B*z  
E =  
    648
```

6. Решить систему линейных уравнений.

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 = b_2 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 = b_3 \end{cases}$$

Пример

$$\begin{cases} 1.5x_1 + 3.6x_2 - 0.5x_3 = 4.6 \\ 0.6x_1 + 5.6x_2 + 10x_3 = 7.6 \\ -0.4x_1 + 5.4x_2 + 8.2x_3 = 5.8 \end{cases}$$

Код программы в MATLAB

```
A=[1.5 3.6 -0.5;0.6 5.6 10;-0.4 5.4 8.2 ];
b=[4.6;7.6;5.8];
x=A\b
x =
    1.2135
    0.8050
    0.2364
```

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ №5

Построение и работа с блочными матрицами

Цель занятия: изучение на практике принципов работы с блочными матрицами в MATLAB.

Задание

1. Ввести четыре квадратные матрицы (A, B, C, D) размерностью два и создать из них блочную матрицу $E = \begin{pmatrix} A & B \\ C & D \end{pmatrix}$

Пример

$$A = \begin{pmatrix} 2 & 3 \\ 4 & 5 \end{pmatrix}, B = \begin{pmatrix} 6 & 7 \\ 8 & 9 \end{pmatrix}, C = \begin{pmatrix} 10 & 11 \\ 12 & 13 \end{pmatrix}, D = \begin{pmatrix} 14 & 15 \\ 16 & 17 \end{pmatrix}$$

Код программы в MATLAB

```
A=[2 3;4 5]
B=[6 7;8 9]
C=[10 11;12 13]
D=[14 15;16 17]
E=[A B;C D]
A =
     2     3
     4     5
B =
     6     7
     8     9
```

```

C =
    10    11
    12    13
D =
    14    15
    16    17
E =
     2     3     6     7
     4     5     8     9
    10    11    14    15
    12    13    16    17

```

2. Составить блочную матрицу $H = \left(\begin{array}{c|c} Z & a \\ \hline b & N \end{array} \right)$, где $Z = \begin{pmatrix} z_{11} & z_{12} \\ z_{21} & z_{22} \end{pmatrix}$, $a = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix}$,

$b = (b_1 \ b_2)$, N - заданное число.

Пример

$$Z = \begin{pmatrix} 2 & 3 \\ 4 & 5 \end{pmatrix}, a = \begin{pmatrix} 6 \\ 8 \end{pmatrix}, b = (-10 \ 15), N=3.6$$

Код программы в MATLAB

```

Z=[2 3;4 5];
a=[6;8];
b=[-10 15];
N=3.6;
H=[Z a;b N]

H =
    2.0000    3.0000    6.0000
    4.0000    5.0000    8.0000
   -10.0000   15.0000    3.6000

```

3. Выделить блоки из полученной матрицы $E = \begin{pmatrix} \boxed{\begin{smallmatrix} \vdots & \vdots \end{smallmatrix}} & \boxed{\begin{smallmatrix} \vdots & \vdots \end{smallmatrix}} \\ \boxed{\begin{smallmatrix} \vdots & \vdots \end{smallmatrix}} & \boxed{\begin{smallmatrix} \vdots & \vdots \end{smallmatrix}} \end{pmatrix}$ и выделить 3-ю строку из матрицы H .

Код программы в MATLAB

```

A=[2 3;4 5];
B=[6 7;8 9];
C=[10 11;12 13];
D=[14 15;16 17];
E=[A B;C D];
Z=[2 3;4 5];
a=[6;8];
b=[-10 15];
N=3.6;

```

```

H=[Z a;b N];
E1=E(1:2,3:4)
E2=E(3:4,1:2)
h=H(3,:)
E1 =
     6     7
     8     9
E2 =
    10    11
    12    13
h =
-10.0000    15.0000     3.6000

```

4. Удалить первую строку матрицы E и 3-ий столбец.

Код программы в MATLAB

```

A=[2 3;4 5];
B=[6 7;8 9];
C=[10 11;12 13];
D=[14 15;16 17];
E=[A B;C D];
E(1,:)=[]
E(:,3)=[]
E =
     4     5     8     9
    10    11    14    15
    12    13    16    17
E =
     4     5     9
    10    11    15
    12    13    17

```

5. Заполнить прямоугольную $n \times l$ и квадратную матрицу $n \times n$ нулями, единицами и случайными числами при помощи индексации и встроенных функций.

Пример

$n=3, l=4$

Код программы в MATLAB

```

A(1:3,1:4)=0
A5(1:3,1:4)=1
A15=rand(3,4)
B(1:3,1:3)=0
B5(1:3,1:3)=1
B15=rand(3,3)
A =
     0     0     0     0

```

```

0      0      0      0
0      0      0      0

A5 =
1      1      1      1
1      1      1      1
1      1      1      1
A15 =
0.0462      0.6948      0.0344      0.7655
0.0971      0.3171      0.4387      0.7952
0.8235      0.9502      0.3816      0.1869
B =
0      0      0
0      0      0
0      0      0
B5 =
1      1      1
1      1      1
1      1      1
B15 =
0.4898      0.7094      0.6797
0.4456      0.7547      0.6551
0.6463      0.2760      0.1626

A=zeros(3,4)
A5=ones(3,4)
A15=rand(3,4)
B=zeros(3)
B5=ones(3)
B15=rand(3)
A =
0      0      0      0
0      0      0      0
0      0      0      0
A5 =
1      1      1      1
1      1      1      1
1      1      1      1
A15 =
0.1190      0.3404      0.7513      0.6991
0.4984      0.5853      0.2551      0.8909
0.9597      0.2238      0.5060      0.9593
B =
0      0      0
0      0      0
0      0      0
B5 =
1      1      1
1      1      1
1      1      1

```



```
B15 =
    0.5472    0.2575    0.8143
    0.1386    0.8407    0.2435
    0.1493    0.2543    0.9293
```

6. Заполнить вектор-строку l случайными числами.

Пример

$l=7$

Код программы в MATLAB

```
a=rand(1,7)
a =
    0.3500    0.1966    0.2511    0.6160    0.4733    0.3517    0.8308
```

7. Создать диагональную матрицу $n \times n$, у которой все внедиагональные элементы равны нулю.

Код программы в MATLAB

```
e=[1;2;3;4;5]
E=diag(e)
E =
     1     0     0     0     0
     0     2     0     0     0
     0     0     3     0     0
     0     0     0     4     0
     0     0     0     0     5
```

8. Заполните и запишите в файлы матрицы:

$$E = \begin{pmatrix} a & 0 & 0 & 0 & 0 & b & b & b & b & b \\ 0 & a & 0 & 0 & 0 & b & b & b & b & b \\ 0 & 0 & a & 0 & 0 & b & b & b & b & b \\ 0 & 0 & 0 & a & 0 & b & b & b & b & b \\ 0 & 0 & 0 & 0 & a & b & b & b & b & b \\ c & c & c & c & c & 1 & 0 & 0 & 0 & 0 \\ c & c & c & c & c & 0 & 1 & 0 & 0 & 0 \\ c & c & c & c & c & 0 & 0 & 1 & 0 & 0 \\ c & c & c & c & c & 0 & 0 & 0 & 1 & 0 \\ c & c & c & c & c & 0 & 0 & 0 & 0 & 1 \end{pmatrix}; H = \begin{pmatrix} e & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & e & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & e & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & e & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & e & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & e & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & e & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & e & 1 \\ 0 & 0 & 0 & 0 & c & 0 & 1 & e \end{pmatrix}$$

Пример

$a = -4, b = 6, c = 4, e = 5$

Код программы в MATLAB

```
E=[-4*eye(5) 6*ones(5);4*ones(5) eye(5)]
save('d:\E.txt','E','-ascii')
H=5*eye(8)+diag(ones(1,7),1)+diag(ones(1,7),-1)
save('d:\H.txt','H','-ascii')
```

```
E =
    -4         0         0         0         0         6         6         6         6         6
         0        -4         0         0         0         6         6         6         6         6
         0         0        -4         0         0         6         6         6         6         6
         0         0         0        -4         0         6         6         6         6         6
         0         0         0         0        -4         6         6         6         6         6
         4         4         4         4         4         1         0         0         0         0
         4         4         4         4         4         0         1         0         0         0
         4         4         4         4         4         0         0         1         0         0
         4         4         4         4         4         0         0         0         1         0
         4         4         4         4         4         0         0         0         0         1

H =
         5         1         0         0         0         0         0         0
         1         5         1         0         0         0         0         0
         0         1         5         1         0         0         0         0
         0         0         1         5         1         0         0         0
         0         0         0         1         5         1         0         0
         0         0         0         0         1         5         1         0
         0         0         0         0         0         1         5         1
         0         0         0         0         0         0         1         5
```

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ №6

Поэлементные операции над матрицами

Цель занятия: изучение на практике поэлементных операций над матрицами.

Задание

1. В матрице A вычислить сумму по столбцам и по строкам; отсортировать элементы этой матрицы в порядке возрастания их столбцов и строк; вычислить максимальные и минимальные элементы в соответствующих столбцах матрицы A и строках.

Пример

$$A = \begin{pmatrix} 3 & -4 & -6 \\ 8 & -2 & 10 \\ 5 & -2 & 4 \end{pmatrix}$$

Код программы в MATLAB

```
A=[3 -4 -6;8 -2 10;5 -2 4];
e=sum(A)%Сумма по столбцам 16 -8 8
e5=sum(A,2)%Сумма по строкам -7 16 7
E=sort(A)%Сортировка в порядке возрастания элементов в столбцах
    %E = 3 -4 -6
        %5 -2 4
        %8 -2 10
E5=sort(A,2)%Сортировка в порядке возрастания элементов в строках
    %E5 = -6 -4 3
        %-2 8 10
        %-2 4 5
amax=max(A)%Максимальные элементы в столбцах 8 -2 10
amin=min(A)%Минимальные элементы в столбцах 3 -4 -6
amax5=max(A,[],2)%Максимальные элементы в строках 3 10 5
amin5=min(A,[],2)%Минимальные элементы в строках -6 -2 -2
```

2. Создать квадратную матрицу A размера N , состоящую из случайных целых чисел от нуля до десяти, вычесть из нее матрицу, состоящую из шестерок. Используя полученную квадратную матрицу A , вычислить следующие величины (нормы матрицы):

$$p = \max_{1 \leq i \leq n} \sum_{k=1}^n |a_{ik}|, q = \max_{1 \leq k \leq n} \sum_{i=1}^n |a_{ik}|, r = \left(\sum_{i,k=1}^n |a_{ik}|^2 \right)^{0.5}, s = \sum_{i,k=1}^n |a_{ik}|.$$

Пример

$N=5$

Код программы в MATLAB

```
N=5;
A=round(10*rand(N))-6*ones(N)
    %A = -3 -1 2 4 2
          %1 4 -3 -1 -3
          %1 -3 -1 -5 2
          %-4 0 1 -5 -4
          %-5 -4 3 -3 3
p=max(sum(abs(A)))% p = 18
q=max(sum(abs(A),2))% q = 18
r=sqrt(sum(sum(abs(A).^2)))% r = 15.3623
s=sum(sum(abs(A)))%s = 68
```

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ №7

Операторы цикла в MATLAB

Цель занятия: изучение на практике реализации средствами MATLAB циклических процедур.

Задание

1. Написать программу для вычисления суммы $S = \sum_{k=1}^N \frac{e^k}{x^k}$; $x \in [a, b]$ с использованием оператора *for*.

Пример

$N=10, a=5, b=15$

Код программы в MATLAB

```
a=5;
b=15;
N=10;
h=(b-a)/N;
s=0;
for i=1:N+1
    x(i)=a+(i-1)*h
    s=s+exp(i)/x(i)
end
%x      =      5      6      7      8      9      10      11      12      13      14      15
%s = 6.6047e+03
```

2. Написать программу для вычисления выражения $Z_{x,y} = \frac{x+y-2}{x+y+4}$;

$x=2,3,\dots,n+1$; $y=1,2,\dots,n$ с использованием оператора *while*. Вывести

график функции $Z_{x,y} = \frac{x+y-2}{x+y+4}$ в виде непрерывной поверхности.

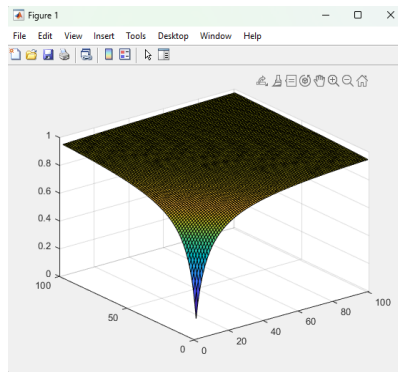
Пример

$n=100$

Код программы в MATLAB

```
n=100;
x=2:1:n+1;%координаты точек по оси Ox
y=1:1:n;%координаты точек по оси Oy
[X,Y]=meshgrid(x,y); %матрицы координат точек по
                     %осям Ox и Oy
while n>=1
    Z=(X+Y-2)./(X+Y+4)
    n=n-1;
end
```

```
hidden off; %скрытые линии рисуются
surf(X,Y,Z); %отображение непрерывной поверхности
```



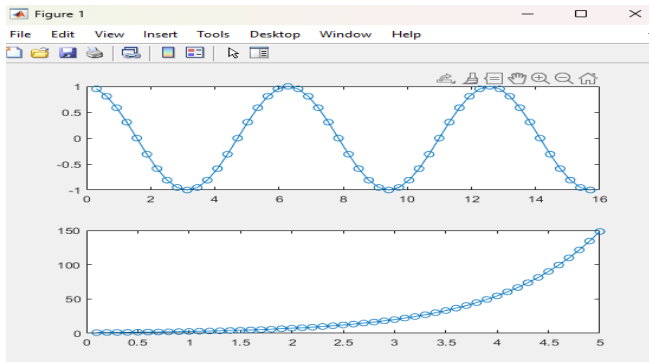
3. Написать программу для нахождения значений функций $\cos(x)$ и e^x путем разложения в ряды: $\sum_{k=0}^{\infty} (-1)^k \frac{x^{2k}}{(2k)!}$; $x \in [a, b]$ и $\sum_{k=0}^{\infty} \frac{x^k}{k!}$; $x \in [a_5, b_5]$ соответственно, пока слагаемые суммы больше 10^{-l} . Вывести графики полученных функций.

Пример

$n=2, l=10, a=0.1\pi, b=5\pi, a_5=0.1, b_5=5$

Код программы в MATLAB

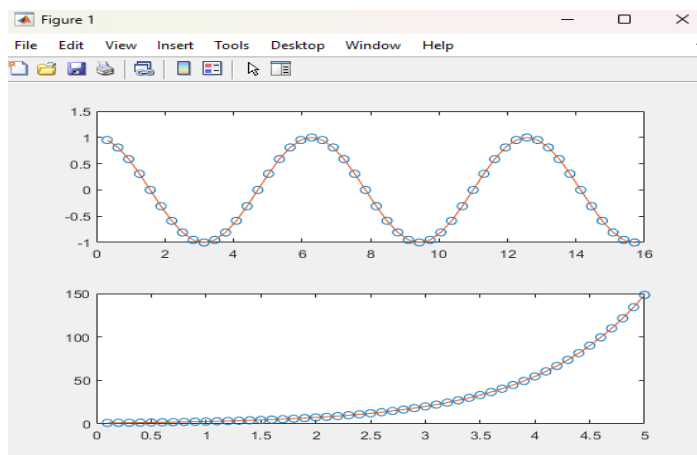
```
for i=1:50
    x(i)=0.1*i*pi
    k=0;
    s=0;
    while abs(x(i).^(2*k)/factorial(2*k))>1.0e-10
        s=s+(-1)^k*x(i).^(2*k)/factorial(2*k)
        k=k+1;
    end
    mycos(i)=s;
end
for i=1:50
    x5(i)=0.1*i
    k=0;
    s5=0;
    while x5(i).^(k)/factorial(k)>1.0e-10
        s5=s5+x5(i).^(k)/factorial(k)
        k=k+1;
    end
    myexp(i)=s5;
end
subplot(2,1,1),plot(x, mycos,'-o')
subplot(2,1,2),plot(x5, myexp,'-o')
```



4. Сравнить полученные результаты, построив графики функций *mycos* и *cos*, *myexp* и *exp*, используя одни и те же значения аргументов.

Код программы в MATLAB

```
for i=1:50
    x(i)=0.1*i*pi
    k=0;
    s=0;
    while abs(x(i).^(2*k)/factorial(2*k))>1.0e-10
        s=s+(-1)^k*x(i).^(2*k)/factorial(2*k)
        k=k+1;
    end
    mycos(i)=s;
end
for i=1:50
    x5(i)=0.1*i
    k=0;
    s5=0;
    while x5(i).^(k)/factorial(k)>1.0e-10
        s5=s5+x5(i).^(k)/factorial(k)
        k=k+1;
    end
    myexp(i)=s5;
end
subplot(2,1,1),plot(x, mycos,'o',x,cos(x),'-')
subplot(2,1,2),plot(x5, myexp,'o',x5,exp(x5),'-')
```



ПРАКТИЧЕСКОЕ ЗАНЯТИЕ №8

Условные операторы

Цель занятия: изучение на практике способов создания гибких разветвляющихся алгоритмов выполнения команд, основанных на применении условных операторов.

Задание

1. Вычислить значение выражения, заданного в варианте, используя условные операторы *if* и *else*.

Пример

Вычислить значение выражения $y(x) = \begin{cases} \sin(x), & x \leq -\pi; \\ x/\pi, & -\pi < x < \pi; \\ \cos(x), & x \geq \pi \end{cases} \quad x \in [-2\pi, 2\pi]$

Код программы в MATLAB

```
x=[-2*pi:2*pi]
N=length(x)
for i=1:N
    if x(i)<=-pi
        y(i)=sin(x(i))
    else if -pi<x(i) & x(i)<pi
        y(i)=x(i)/pi
    else if x(i)>=pi
        y(i)=cos(x(i))
    end
end
end
end
end
%N=13
x=
-6.2831 -5.2831 -4.2831 -3.2831 -2.2831 -1.2831 -0.2831 0.7168
1.7168 2.7168 3.7168 4.7168 5.7168
y=
2.4492e-16 0.8414 0.9092 0.1411 -0.7267 -0.4084 -0.0901
0.2281 0.5464 0.8647 -0.8390 0.0044 0.8438
```

2. Вычислить значение выражения, заданного в варианте, используя условные операторы *switch*, *case* и *otherwise*.

Пример

Вычислить значение выражения $y(x) = \begin{cases} \sin(x), & x = -\pi; \\ x/\pi, & x = \pi; \quad x = -\pi, \pi, 2\pi \\ \cos(x), & x \neq -\pi \wedge x \neq \pi \end{cases}$

Код программы в MATLAB

```
x=[-pi pi 2*pi]
N=length(x)
for i=1:N
switch(x(i))
    case -pi
        y(i)=sin(x(i))
    case pi
        y(i)=x(i)/pi
    otherwise
        y(i)=cos(x(i))
end
end
x =
    -3.1416     3.1416     6.2832
N =
     3
y =
    -0.0000     1.0000     1.0000
```

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ №9

Вывод графиков, диаграмм и гистограмм

Цель занятия: изучение на практике способов вывода графиков, диаграмм и гистограмм в MATLAB.

Задание

1. Отобразить функцию $f(x)$ на отрезке $[a, b]$, заданную в варианте, в виде столбчатой диаграммы без промежутков.

Пример

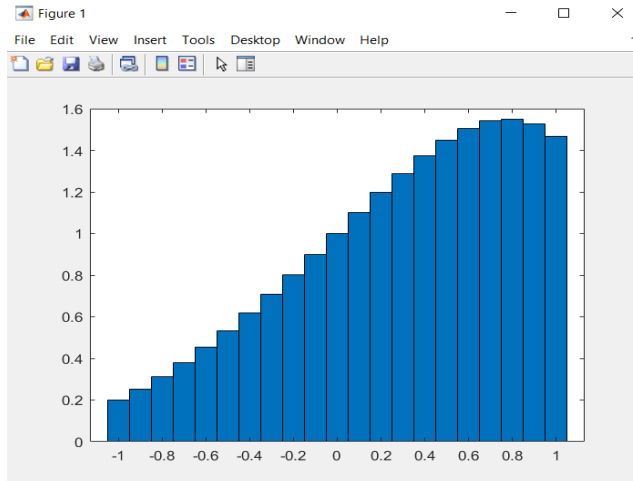
$$f(x) = \cos(x)e^x; \quad x \in [-1, 1]$$

Код программы в MATLAB

```
x=[-1:0.1:1];
y=cos(x).*exp(x);
```



```
bar(x,y,1.0)
```



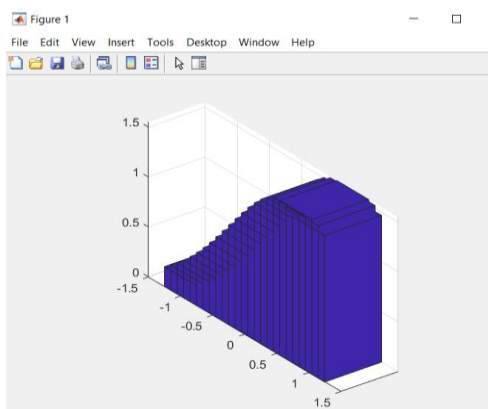
2. Отобразить функцию $f(x)$ на отрезке $[a, b]$, заданную в варианте, в виде объемной столбчатой диаграммы без промежутков.

Пример

$$f(x) = \cos(x)e^x; \quad x \in [-1, 1]$$

Код программы в MATLAB

```
x=[-1:0.1:1];  
y=cos(x).*exp(x);  
bar3(x,y,5.0)
```



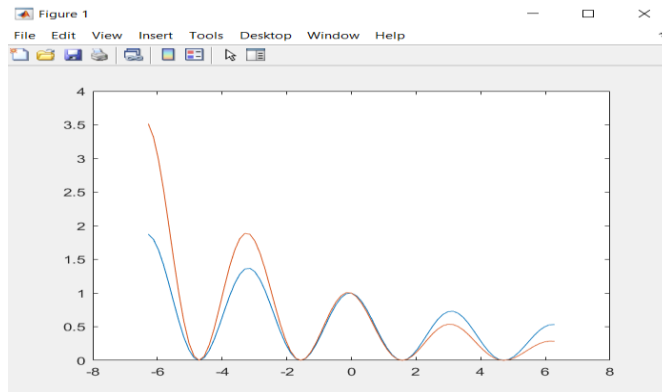
3. Построить графики функций $f(x)$ и $\varphi(x)$ на отрезке $[a, b]$, заданную в варианте. Сгенерировать вектор-строку значений аргумента x и вектор-строку f и φ , содержащих значения функций.

Пример

$$f(x) = \cos^2(x)e^{-0.1x}; \quad \varphi(x) = \cos^2(x)e^{-0.2x}; \quad x \in [-2\pi, 2\pi]$$

Код программы в MATLAB

```
x=[-2*pi:pi/20:2*pi];  
f=cos(x).^2.*exp(-0.1*x);  
f5=cos(x).^2.*exp(-0.2*x);  
plot(x,f,x,f5)
```



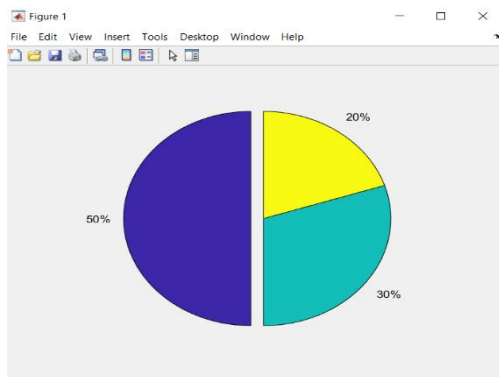
4. Написать программу построения диаграммы с отдельным сектором, соответствующим максимальному значению среди элементов вектора $data=[a_1 \ a_2 \ a_3]$, автоматически создав вспомогательный вектор.

Пример

$data=[25 \ 15 \ 10]$

Код программы в MATLAB

```
data=[25 15 10]  
parts=zeros(size(data));  
[mx,ind]=max(data);  
parts(ind)=1;  
pie(data,parts)
```



5. Написать программу построения объемной диаграммы с отдельным сектором, соответствующим максимальному значению среди

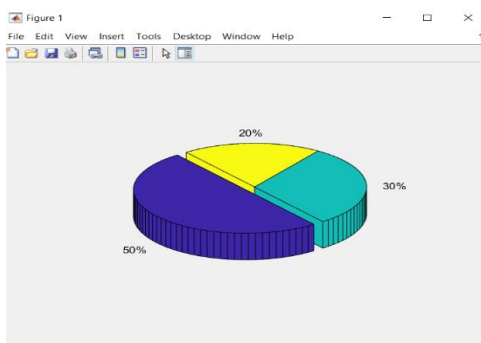
элементов вектора $data=[a_1 \ a_2 \ a_3]$, автоматически создав вспомогательный вектор.

Пример

$$data=[25 \ 15 \ 10]$$

Код программы в MATLAB

```
data=[25 15 10]
parts=zeros(size(data));
[mx,ind]=max(data);
parts(ind)=1;
pie3(data,parts)
```



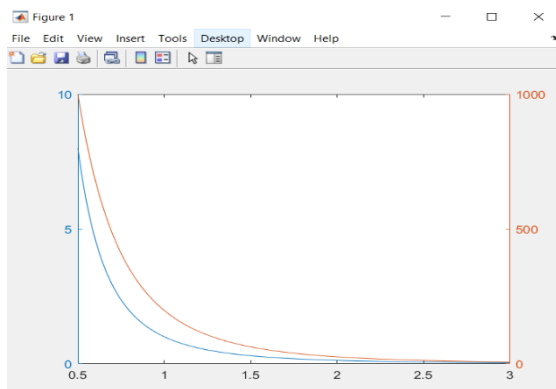
6. Вывести две функции $f(x)$ и $\varphi(x)$ на отрезке $[a, b]$ в окно с двумя вертикальными осями, имеющими подходящий масштаб.

Пример

$$f(x) = \frac{1}{x^3}; \quad \varphi(x) = 1000(x+0.5)^{-4}; x \in [0.5, 3]$$

Код программы в MATLAB

```
x=[0.5:0.01:3];
f=x.^-3;
f5=1000*(x+0.5).^-4;
plotyy(x,f,x,f5)
```



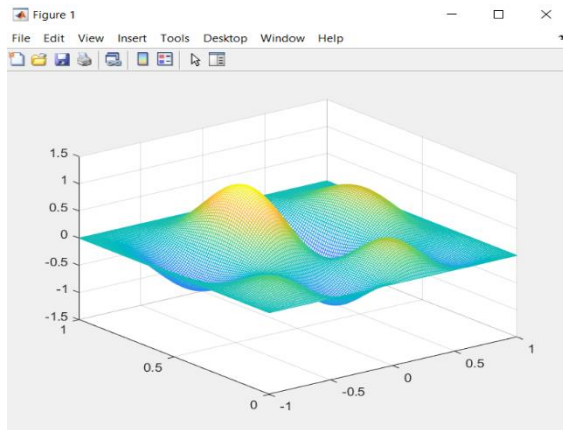
7. Построить график функции двух переменных $z = f(x, y)$ на прямоугольной области определения $x \in [a, b]$, $y \in [a_5, b_5]$ в виде сетки.

Пример

$$z(x, y) = 5 \sin(2\pi x) \cos(1.5\pi y) (1 - x^2) y (1 - y), \quad x \in [-1, 1], \quad y \in [0, 1]$$

Код программы в MATLAB

```
x=[-1:0.01:1];  
y=[0:0.01:1];  
[X,Y]=meshgrid(x,y);  
Z=5*sin(2*pi*X).*cos(1.5*pi*Y).*(1-X.^2).*Y.*(1-Y);  
mesh(X,Y,Z)
```



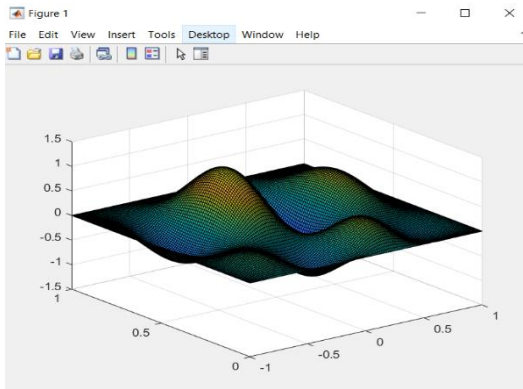
8. Построить график функции двух переменных $z = f(x, y)$ на прямоугольной области определения $x \in [a, b]$, $y \in [a_5, b_5]$ в виде непрерывной поверхности.

Пример

$$z(x, y) = 5 \sin(2\pi x) \cos(1.5\pi y) (1 - x^2) y (1 - y), \quad x \in [-1, 1], \quad y \in [0, 1]$$

Код программы в MATLAB

```
x=[-1:0.01:1];  
y=[0:0.01:1];  
[X,Y]=meshgrid(x,y);  
Z=5*sin(2*pi*X).*cos(1.5*pi*Y).*(1-X.^2).*Y.*(1-Y);  
surf(X,Y,Z)
```



ПРАКТИЧЕСКОЕ ЗАНЯТИЕ №10

Работа с передаточными функциями

Цель занятия: изучение на практике основных возможностей работы с передаточными функциями в MATLAB.

Задание

1. Построить передаточную функцию системы, заданную уравнением

$$W(p) = \frac{A(p)}{B(p)}, \text{ где } A(p), B(p) - \text{полиномы, заданные в варианте.}$$

Пример

$$A(p) = 5p + 1.5, \quad B(p) = 5p^2 + 4.5p + 6$$

Код программы в MATLAB

```
A=[5 1.5];
B=[5 4.5 6];
w=tf(A, B, 'variable','p')
w =
      5 p + 1.5
-----
    5 p^2 + 4.5 p + 6
Continuous-time transfer function.
```

2. Преобразовать построенную передаточную функцию в нули/полюсы/коэффициент усиления.

Код программы в MATLAB

```
w=tf([5 1.5], [5 4.5 6], 'variable','p')
zpk(w)
w =
      5 p + 1.5
-----
    5 p^2 + 4.5 p + 6
Continuous-time transfer function.
```

```
ans =
      (p+0.3)
      -----
      (p^2 + 0.9p + 1.2)
Continuous-time zero/pole/gain model.
```

3. Построить передаточную функцию процесса и преобразовать ее в пространство состояний.

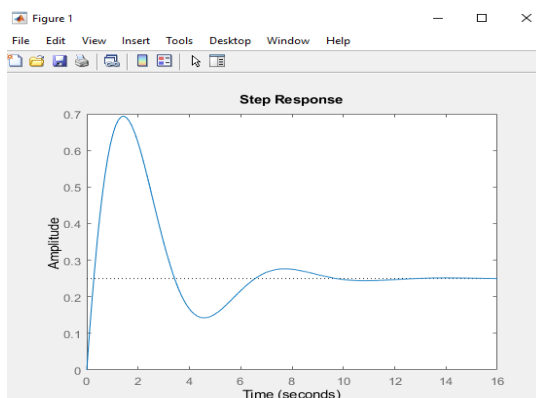
Код программы в MATLAB

```
A=[5 1.5];
B=[5 4.5 6];
w=tf(A, B, 'variable','p');
ss(w)
ans =
      A =
           x1      x2
x1  -0.9   -1.2
x2      1      0
      B =
           u1
x1      1
x2      0
      C =
           x1      x2
y1      1   0.3
      D =
           u1
y1      0
Continuous-time state-space model.
```

4. Осуществить построение переходного процесса для сформированной передаточной функции в графическом окне.

Код программы в MATLAB

```
w=tf([5 1.5], [5 4.5 6], 'variable','p');
step(w)
```



5. Осуществить вычисление показателей демпфирования и собственных частот системы.

Код программы в MATLAB

```
w=tf([5 1.5], [5 4.5 6], 'variable', 'p');
damp(w)
```

Pole	Damping	Frequency (rad/seconds)	Time Constant (seconds)
-4.50e-01 + 9.99e-01i	4.11e-01	1.10e+00	2.22e+00
-4.50e-01 - 9.99e-01i	4.11e-01	1.10e+00	2.22e+00

6. Осуществить определение статического коэффициента передачи сформированной передаточной функции.

Код программы в MATLAB

```
w=tf([5 1.5], [5 4.5 6], 'variable', 'p');  
dcgain(w)  
ans =  
    0.2500
```

7. Осуществить определение корней сформированной передаточной функции.

Код программы в MATLAB

```
w=tf([5 1.5], [5 4.5 6], 'variable', 'p');
eig(w)
pole(w)
ans =
    -0.4500 + 0.9987i
    -0.4500 - 0.9987i
ans =
    -0.4500 + 0.9987i
    -0.4500 - 0.9987i
```

8. Осуществить определение полюса сформированной передаточной функции.

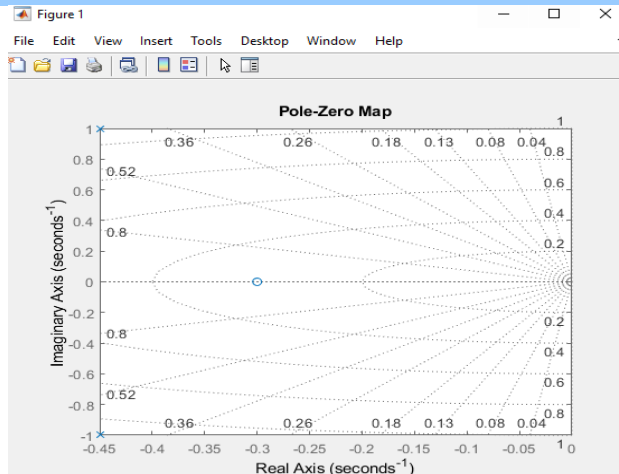
Код программы в MATLAB

```
w=tf([5 1.5], [5 4.5 6],'variable','p');
tzero(w)
ans =
    -0.3000
```

9. Осуществить определение нулей и полюсов сформированной передаточной функции.

Код программы в MATLAB

```
w=tf([5 1.5], [5 4.5 6], 'variable', 'p');
pzmap(w)
grid
```



10. Построить передаточную функцию системы, заданную передаточными функциями элементов системы: W , W_1 , W_2 , а также типом соединений между ними:

- все элементы системы соединены последовательно;
- все элементы системы соединены параллельно;
- первый и второй элементы соединены последовательно, третий элемент включен в обратную связь (положительную, отрицательную).

Пример

$$W = \frac{5p+1}{p^2+p+1}, W_1 = \frac{4p+1}{5p^2+p+1}, W_2 = \frac{6p+1}{4p^2+p+1}$$

Код программы в MATLAB

```
w = tf([5 1], [1 1 1], 'variable', 'p')
w1 = tf([4 1], [5 1 1], 'variable', 'p')
w2 = tf([6 1], [4 1 1], 'variable', 'p')
sys1=series(series(w,w1),w2) %последовательное соединение трех
                             %элементов
```



```

sys2=parallel(parallel(w,w1),w2) %параллельное соединение трех
                                %элементов
sys3=feedback(series(w,w1),w2) %с отрицательной обратной связью
sys4=feedback(series(w,w1),w2,+1) % с положительной обратной
                                %связью

w =
      5 p + 1
      -----
      p^2 + p + 1
Continuous-time transfer function.
w1 =
      4 p + 1
      -----
      5 p^2 + p + 1
Continuous-time transfer function.
w2 =
      6 p + 1
      -----
      4 p^2 + p + 1
Continuous-time transfer function.
sys1 =
      120 p^3 + 74 p^2 + 15 p + 1
      -----
      20 p^6 + 29 p^5 + 39 p^4 + 21 p^3 + 13 p^2 + 3 p + 1
Continuous-time transfer function.
sys2 =
      146 p^5 + 130 p^4 + 136 p^3 + 53 p^2 + 21 p + 3
      -----
      20 p^6 + 29 p^5 + 39 p^4 + 21 p^3 + 13 p^2 + 3 p + 1
Continuous-time transfer function.
sys3 =
      80 p^4 + 56 p^3 + 33 p^2 + 10 p + 1
      -----
      20 p^6 + 29 p^5 + 39 p^4 + 141 p^3 + 87 p^2 + 18 p + 2
Continuous-time transfer function.
sys4 =
      80 p^4 + 56 p^3 + 33 p^2 + 10 p + 1
      -----
      20 p^6 + 29 p^5 + 39 p^4 - 99 p^3 - 61 p^2 - 12 p
Continuous-time transfer function.

```

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ №1

Работа с функциями извлечения данных из модели и анализа свойств модели

Цель занятия: изучение на практике функций извлечения данных из модели и анализа свойств модели в MATLAB.

Задание

1. Осуществить извлечение параметров заданной передаточной функции $W(p) = \frac{A(p)}{B(p)}$, где $A(p), B(p)$ - полиномы, заданные в варианте.

Пример

$$A(p) = 5p + 1, \quad B(p) = 5p^2 + 4p + 1$$

Код программы в MATLAB

```
w = tf([5 1], [5 4 1], 'variable', 'p');  
tfdata(w)  
ans =  
    1×1 cell array  
    {[0 5 1]}
```

2. Осуществить извлечение нулей/полюсов/коэффициентов усиления системы, заданной передаточной функцией.

Код программы в MATLAB

```
w=tf([5 1], [5 4 1], 'variable', 'p');  
w1=zpk(w)  
zpkdata(w1)  
w1 =  
      (p+0.2)  
-----  
      (p^2 + 0.8p + 0.2)  
Continuous-time zero/pole/gain model.  
ans =  
    1×1 cell array  
    {[ -0.2000]}
```

3. Осуществить извлечение матриц пространства состояния системы, заданной передаточной функцией.

Код программы в MATLAB

```
w = tf([5 1], [5 4 1], 'variable', 'p');  
w1=ss(w)  
ssdata(w1)  
w1 =  
    A =  
      x1      x2  
x1  -0.8   -0.4  
x2   0.5     0  
    B =  
      u1  
x1     1
```

```

x2    0
C =
      x1    x2
y1     1    0.4
D =
      u1
y1     0
Continuous-time state-space model.
ans =
    -0.8000    -0.4000
     0.5000         0

```

4. Осуществить получение свойств LTI- модели системы, заданной передаточной функцией.

```

w = tf([5 1], [5 4 1], 'variable', 'p');
get(w)
Numerator: {[0 5 1]}
Denominator: {[5 4 1]}
Variable: 'p'
IODelay: 0
InputDelay: 0
OutputDelay: 0
Ts: 0
TimeUnit: 'seconds'
InputName: {''}
InputUnit: {''}
InputGroup: [1x1 struct]
OutputName: {''}
OutputUnit: {''}
OutputGroup: [1x1 struct]
Notes: [0x1 string]
UserData: []
Name: ''
SamplingGrid: [1x1 struct]

```

5. Осуществить извлечение данных числителя и знаменателя и проверку результатов вычислений и преобразований моделей по корням числителя и знаменателя передаточных функций.

Код программы в MATLAB

```

w = tf([5 1], [5 4 1], 'variable', 'p')
[n,d]=tfdata(w,'v')
roots([5 1])
roots([5 4 1])
w =
      5 p + 1
-----

```

```

5 p^2 + 4 p + 1
Continuous-time transfer function.
n =
    0     5     1
d =
    5     4     1
ans =
    -0.2000
ans =
    -0.4000 + 0.2000i
    -0.4000 - 0.2000i

```

6. Осуществить анализ свойств модели, заданной передаточной функцией, различными функциями: *step*, *impulse*, *bode*, *margin*, *rlocus*.

Пример

$$W(p) = \frac{A(p)}{B(p)} = \frac{5p+1}{5p^2+4p+1}$$

Коды программ в MATLAB

```

w = tf([5 1], [5 4 1], 'variable', 'p')
step(w);

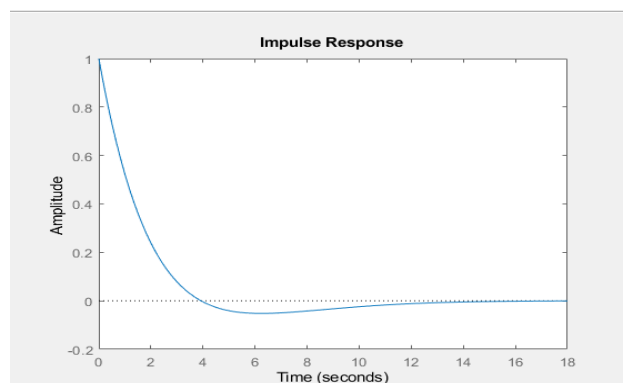
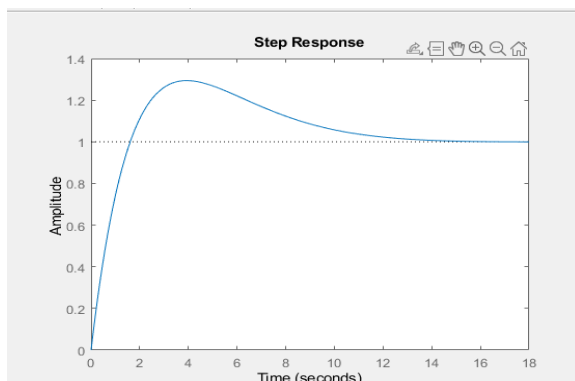
w = tf([5 1], [5 4 1], 'variable', 'p')
impulse(w);

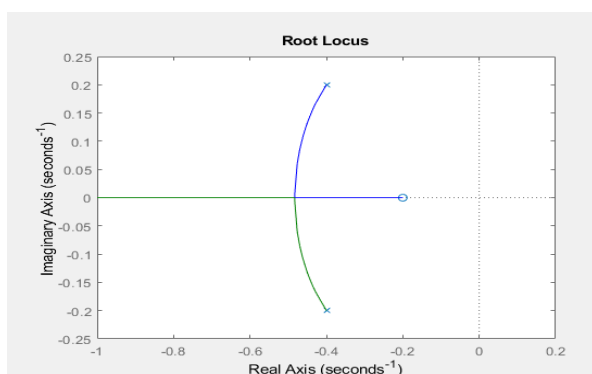
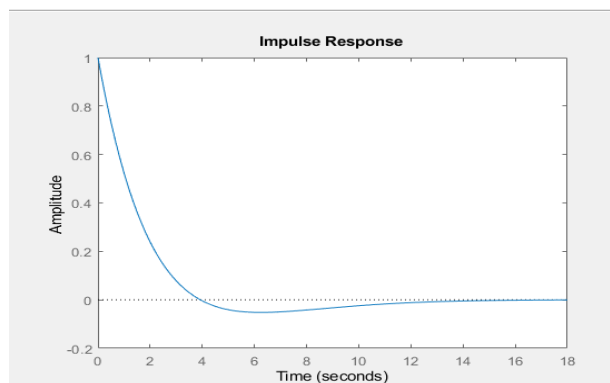
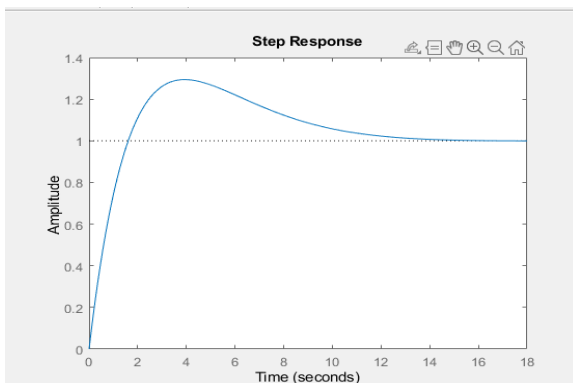
w = tf([5 1], [5 4 1], 'variable', 'p')
bode(w);

w = tf([5 1], [5 4 1], 'variable', 'p')
margin(w);

w = tf([5 1], [5 4 1], 'variable', 'p')
rlocus(w);

```





ПРАКТИЧЕСКОЕ ЗАНЯТИЕ №12

Интерфейс браузера библиотек SIMULINK

Цель занятия: изучение на практике основных приемов работы с библиотекой SIMULINK.

Задание

1. Открыть окно браузера библиотек SIMULINK, выполнив последовательность команд **Simulink - Blank Model** и нажав кнопку **Library Browser** на открывшейся панели редактирования модели.
2. Изучить элементы окна браузера библиотеки Simulink Library Browser: панель инструментов, окно комментария, список разделов библиотеки, окно содержимого раздела библиотеки (рис. 12.1).
3. Изучить основные блоки библиотеки Simulink Library, используя полный перечень блоков, выполнив последовательность команд **Help - Blocks - Block Libraries**.
4. Изучить настройки параметров блоков, приведенных в варианте задания.

5. Открыть окно редактора новой модели и перетащить блоки из библиотеки SIMULINK, указанные в варианте задания, в окно модели.

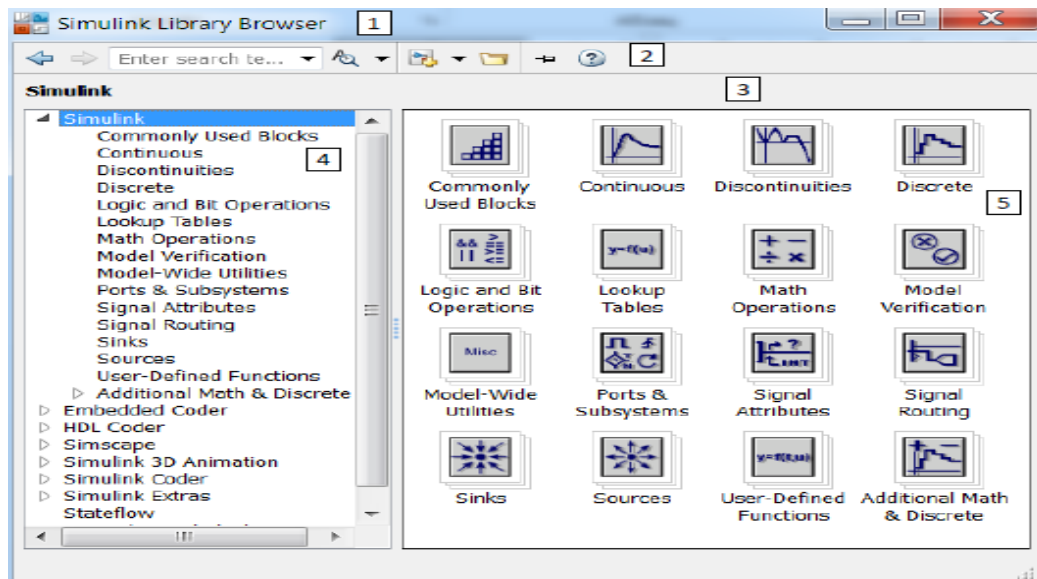


Рисунок 12.1- Окно браузера библиотеки SIMULINK

1- заголовок, с названием окна - Simulink Library Browser; 2- панель инструментов; 3- окно комментария; 4- список разделов библиотеки; 5- окно содержимого раздела библиотеки

6. Произвести настройку параметров блоков, помещенных в окно модели.

Пример

Необходимо вывести в окно следующие блоки библиотеки SIMULINK:

- блок **Step** с параметрами: **Step time**: 0.05, **Final value**: 1, **Sample time**: 0.1;
- блок **Sum** с параметрами: **Output minimum**: [1], **Output maximum**: [1] и действием «+-»;
- блок **Saturation** с параметрами: **Upper limit**: 5, **Lower limit**: -0.5;
- блок **Transfer Fcn** с параметрами: **Numerator coefficients**: [1,8,2], **Denominator coefficients**: [336,145,25,1]. Оставшиеся параметры не изменяются;
- ПИД-регулятор с параметрами: **Proportional (P)**: 2, **Integral (I)**: 0.15, **Derivative (D)**: 2. Оставшиеся параметры не изменяются.

Порядок выполнения задания

1. Открыть окно редактора новой модели в SIMULINK и окно браузера библиотек SIMULINK.

2. Перетащить блок **Step** из раздела библиотеки **Sources** в окно редактора модели SIMULINK. Дважды щелкнув левой кнопкой мыши на блоке **Step**, открыть окно редактирования значений параметров блока и установить следующие значения: **Step time**= 0.05, **Final value**= 1, **Sample time**: 0.1. После этого необходимо нажать кнопку **Apply**, затем **Ok**.

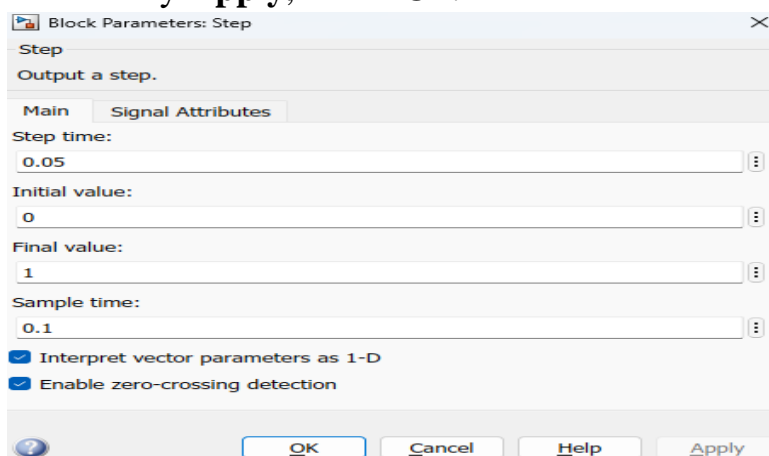


Рисунок 12.2- Окно настройки параметров блока **Step**

3. Перетащить блок **Sum** из раздела библиотеки **Math Operations** в окно редактора модели SIMULINK. Дважды щелкнув левой кнопкой мыши на блоке **Sum** открыть окно редактирования значений параметров блока и в опции **Signal Attributes** установить значения **Output minimum**: [1], **Output maximum**: [1], в опции **Main** - **List of signs** установить последовательность символов «|+-», нажать кнопку **Apply**, затем **Ok**.

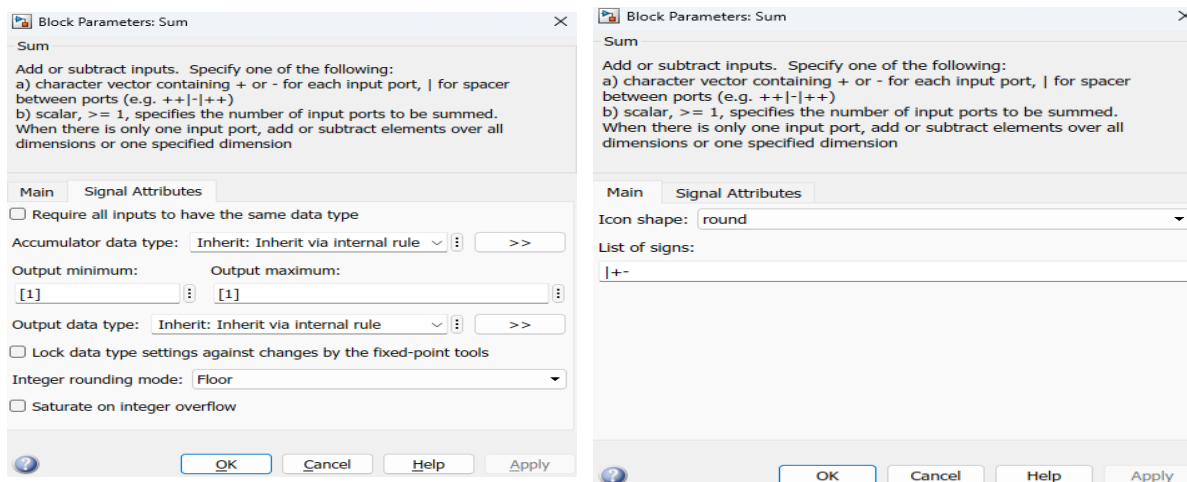


Рисунок 12.3- Окна настройки параметров блока **Sum**

4. Перетащить блок **Saturation** из раздела библиотеки **Commonly Used Bloks** в окно редактора модели SIMULINK. Дважды щелкнув левой кнопкой мыши на блоке **Saturation** открыть окно редактирования значений параметров блока и в опции **Main** установить значения: **Upper limit:** 5, **Lower limit:** -0.5, в опции **Signal Attributes** установить значения **Output minimum:** [1], **Output maximum:** [1], нажать кнопку **Apply**, затем **Ok**.

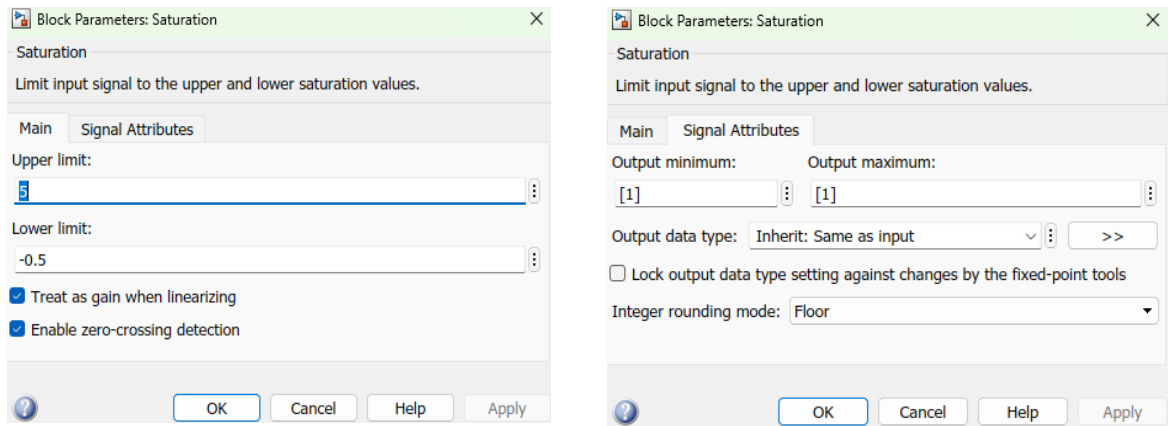


Рисунок 12.4- Окна настройки параметров блока **Saturation**

5. Перетащить блок **Transfer Fcn** из раздела библиотеки **Continuous** в окно редактора модели SIMULINK. Дважды щелкнув левой кнопкой мыши на блоке **Transfer Fcn** открыть окно редактирования значений параметров блока и установить значения **Numerator coefficients:** [1 8 2], **Denominator coefficients:** [336 145 25 1], нажать кнопку **Apply**, затем **Ok**. Оставшиеся параметры не изменяются.

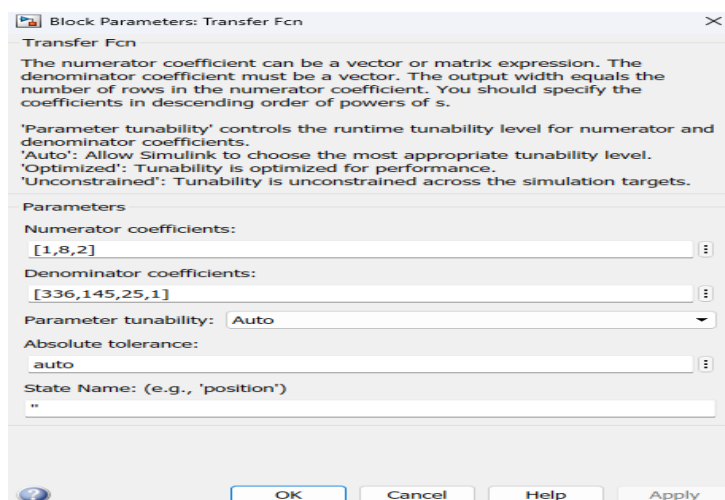


Рисунок 12.5- Окна настройки параметров блока **Transfer Fcn**

6. Перетащить блок **PID Controller** из раздела библиотеки **Commonly Used Bloks** в окно редактора модели SIMULINK. Дважды щелкнув левой кнопкой мыши на блоке **PID Controller** открыть окно редактирования значений параметров блока и установить значения **Proportional (P): 2**, **Integral (I): 0.15**, **Derivative (D): 2**, нажать кнопку **Apply**, затем **Ok**. Оставшиеся параметры не изменяются.

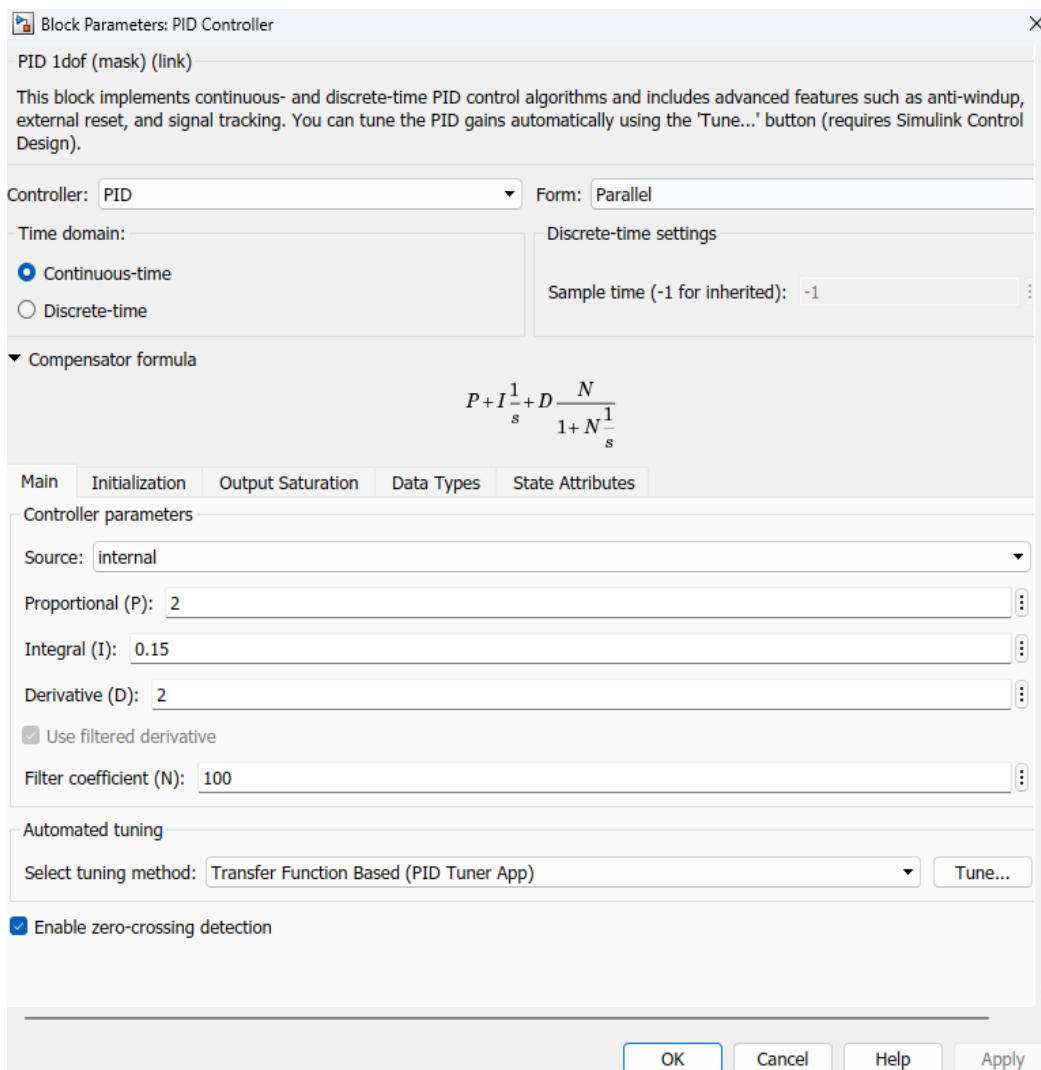


Рисунок 12.6- Окно настройки параметров блока **PID Controller**

Окончательно в окне редактора модели получим заданный набор блоков (рис. 13.7). Соединив блоки в необходимом порядке можно получить структурную модель системы.

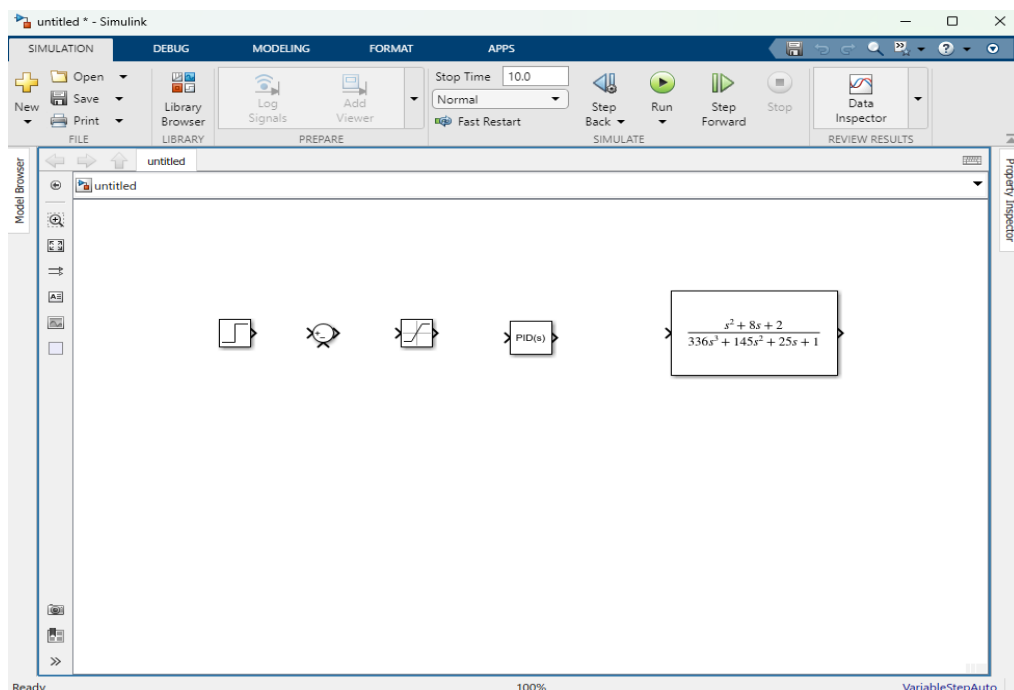


Рисунок 12.7- Окно редактирования модели в SIMULINK

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ №13

Создание моделей в SIMULINK

Цель занятия: изучение на практике основных приемов подготовки и редактирования моделей в SIMULINK.

Задание

1. Открыть окно новой модели SIMULINK.
2. Расположить окно новой модели рядом с окном браузера библиотеки SIMULINK.
3. Из указанных в заданном варианте разделов библиотеки SIMULINK перенести в окно модели требуемые блоки системы.
4. Выполнить соединение между блоками, указанное в варианте.
5. Произвести настройку параметров блоков, указанных в варианте.
6. Произвести установку указанного в варианте времени моделирования.
7. Запустим модель на исполнение, нажав кнопку **StartSimulation** в панели инструментов окна модели.

8. Получить и зафиксировать показания осциллографа.

Пример

Необходимо подготовить, отредактировать и запустить структурную модель системы, схема которой приведена на рисунке:

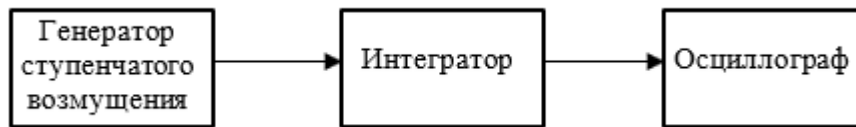


Рисунок 13.1- Структурная схема системы

Порядок выполнения задания

1. Открываем окно новой модели в SIMULINK: **Simulink -Blank Model**.
2. Входим в браузер библиотеки SIMULINK: **Library Browser** и располагаем окна новой модели и браузера библиотеки SIMULINK рядом.

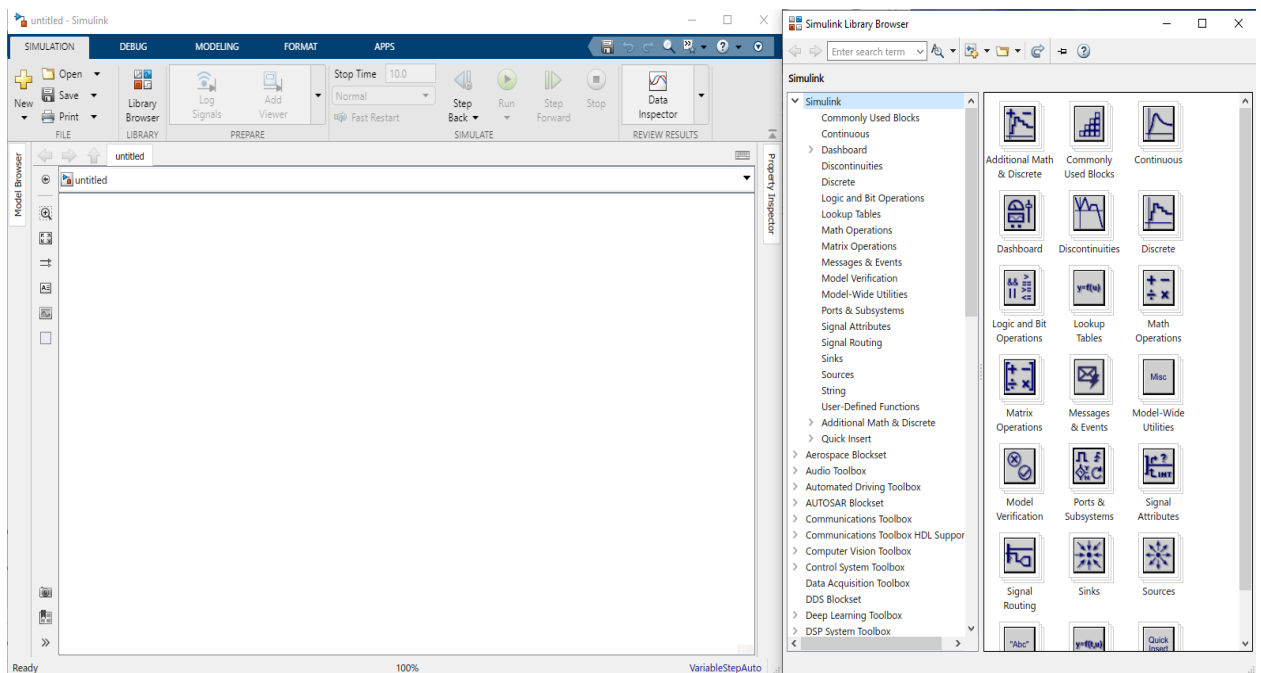


Рисунок 13.2- Окно редактора модели и браузера библиотек SIMULINK

3. 1). Открываем раздел библиотеки **Simulink-Sources** и из открывшегося окна перетаскиваем блок **Step** в окно новой модели.
- 2). Открываем раздел библиотеки **Simulink-Continuous** и из открывшегося окна перетаскиваем блок **Integrator** в окно новой модели.

- 3). Открываем раздел библиотеки **Simulink-Sinks** и из открывшегося окна перетаскиваем блок **Scope** в окно новой модели.
4. Производим соединение блоков и получаем структурную модель системы.

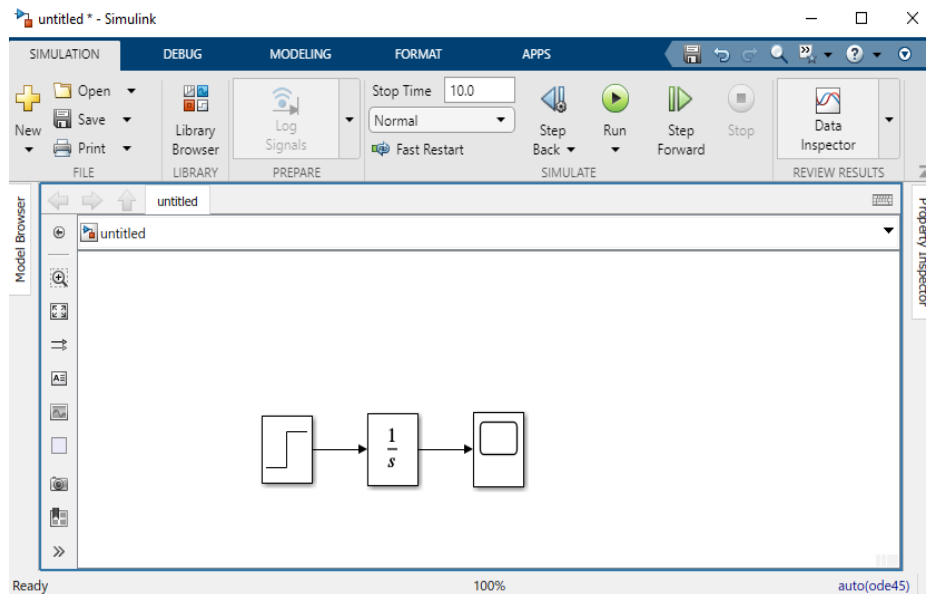


Рисунок 13.3- Структурная модель в окне моделей SIMULINK

5. Осуществляем настройку параметров блока **Step** структурной модели. Настройки двух других блоков остаются первоначальными.

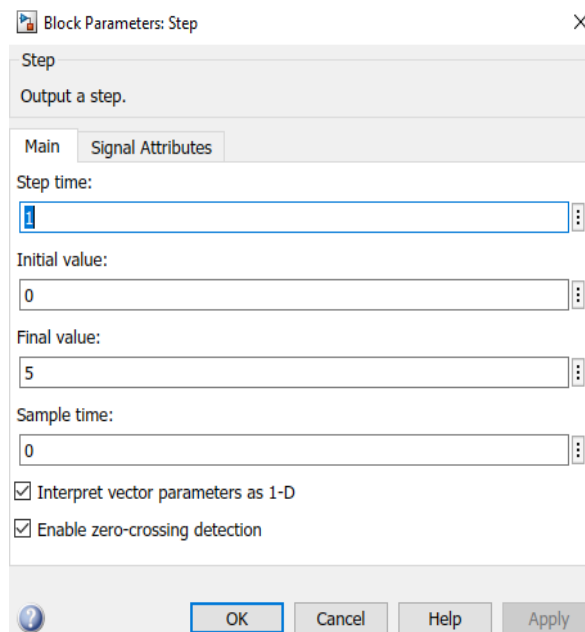


Рисунок 13.4- Окно настройки параметров блока **Step**

6. Устанавливаем время в окне **Stop Time** равным 50.0 секунд.
7. Нажатием кнопки **Run** запускаем работу модели.

Открываем осциллограф и копируем график в документ.

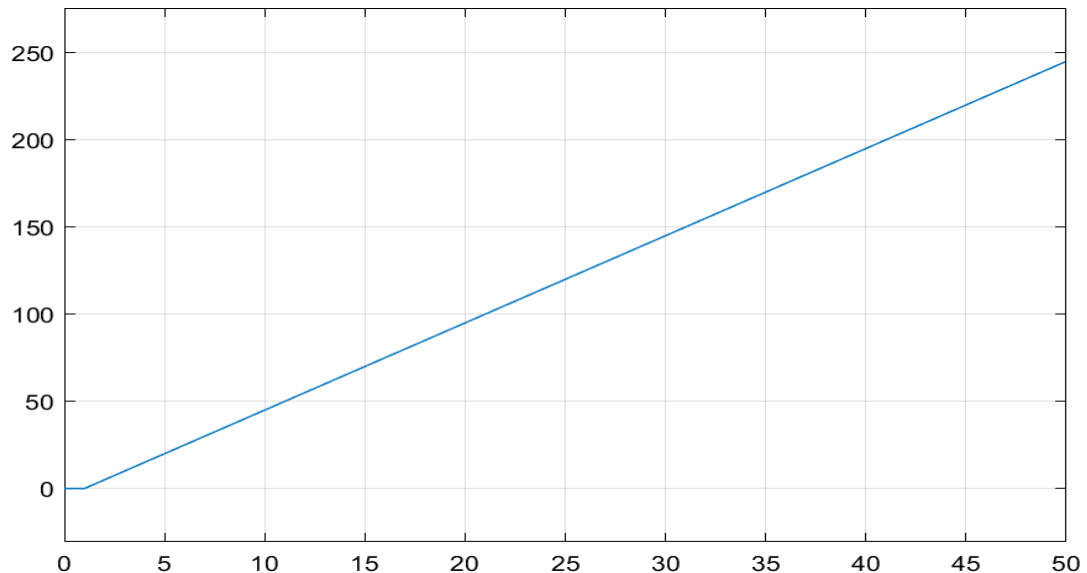


Рисунок 13.5- Окно осциллографа с графиком переходного процесса на выходе блока **Integrator**

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ №14

Анализ моделей в SIMULINK

Цель занятия: изучение на практике основных приемов анализа структурных моделей в SIMULINK.

Задание

1. Построить структурную модель в окне редактора моделей SIMULINK, заданную в варианте.
2. Осуществить подключение линейных аналитических точек **Input Point** и **Output Point**.
3. В меню **Apps** выбрать пункт **Model Linearizer**, в котором запустить функцию **Linear Analysis**.
4. В открывшемся окне в меню **Linearize** выбирать команду, соответствующую необходимому типу характеристики.

Пример

Необходимо произвести анализ характеристик системы, структурная модель которой приведена на рисунке 14.1, с использованием команд: **Step**, **Impulse**, **Bode**, **Nyquist**, **Nichols**, **Singular Values**, **Pole-Zero Map**.

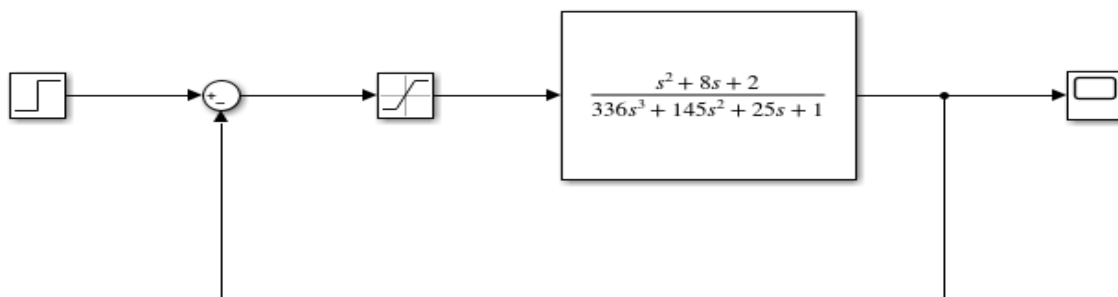


Рисунок 14.1- Структурная модель

Параметры блоков:

- блок **Step** с параметрами: **Step time**: 0.05, **Final value**: 1, **Sample time**: 0.1;
- блок **Sum** с параметрами: **Output minimum**: [1], **Output maximum**: [1] и действием «+»-»;
- блок **Saturation** с параметрами: **Upper limit**: 5, **Lower limit**: -0.5;
- блок **Transfer Fcn** с параметрами: **Numerator coefficients**: [1,8,2], **Denominator coefficients**: [336,145,25,1]. Оставшиеся параметры не изменяются;

ПИД-регулятор с параметрами: **Proportional (P)**: 2, **Integral (I)**: 0.15, **Derivative (D)**: 2.

Порядок выполнения задания

1. Собрать структурную модель, приведенную на рис. 14.1, в соответствии с методикой, описанной в практическом занятии №13, в окне редактирования моделей SIMULINK.
2. Произвести настройку параметров блоков, входящих в структурную модель, в соответствии с методикой, описанной в практическом занятии №12.
3. Осуществить подключение линейных аналитических точек **Input Point** и **Output Point**, как показано на рис. 14.2.

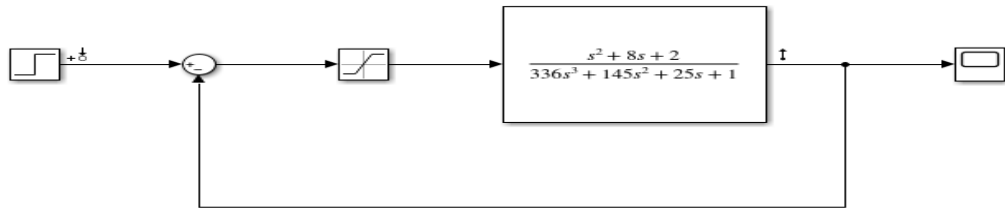


Рисунок 14.2- Схема подключения линейных аналитических точек **Input Point** и **Output Point**

4. Выбрать пункт **Model Linearizer** в меню **Apps** и запустить функцию **Linear Analysis**.
5. Выбирая поочередно команды в меню **Linearize**, соответствующие необходимому типу характеристик, получить следующие графики:
 - график переходного процесса при единичном ступенчатом воздействии;
 - график переходного процесса при импульсном воздействии;
 - логарифмические частотные характеристики ЛАЧХ и ЛФЧХ при гармоническом воздействии;
 - диаграмма Найквиста;
 - частотная характеристика Николса;
 - сингулярные числа;
 - диаграмма распределения полюсов-нулей передаточной функции модели.

Применительно к структурной модели, приведенной на рис. 14.2, перечисленные графики приведены на рис. 14.3-14.10.

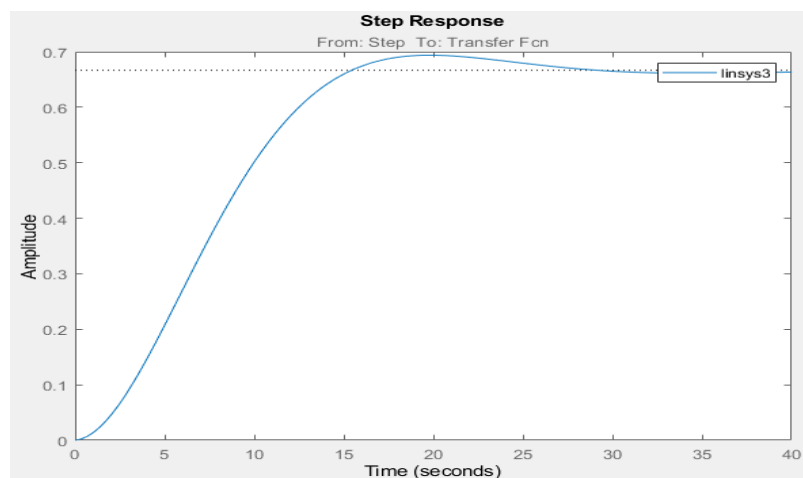


Рисунок 14.3- График переходного процесса при единичном ступенчатом воздействии

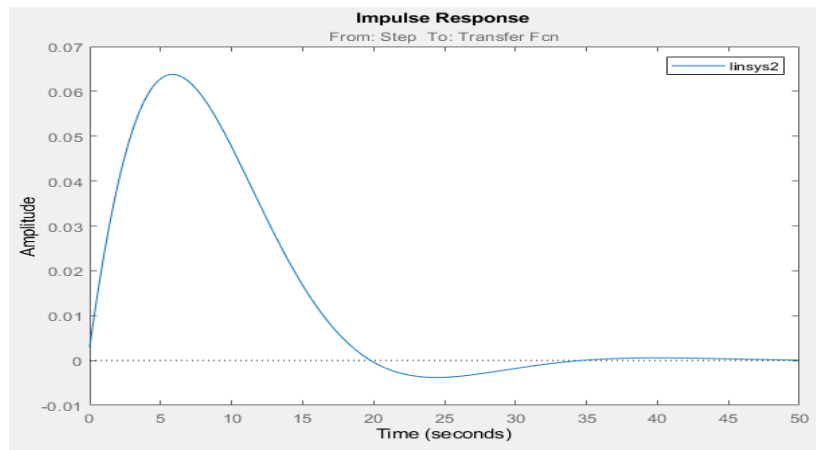


Рисунок 14.4- График переходного процесса при импульсном воздействии

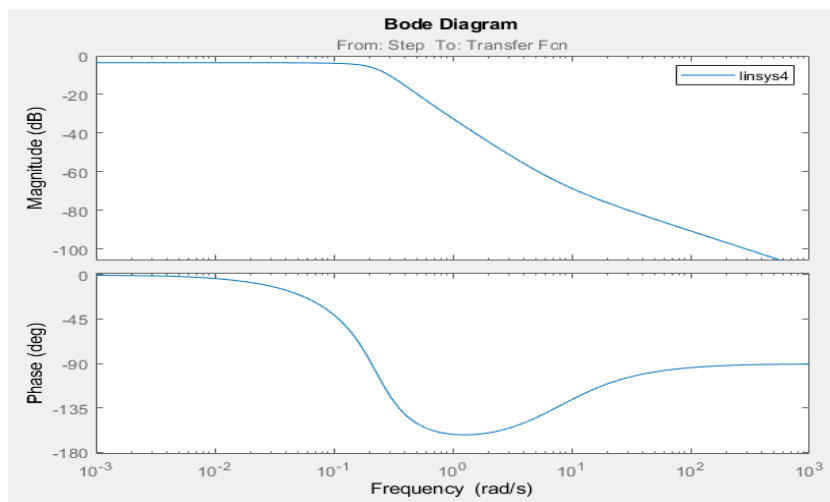


Рисунок 14.5- Логарифмические частотные характеристики ЛАЧХ и ЛФЧХ при гармоническом воздействии

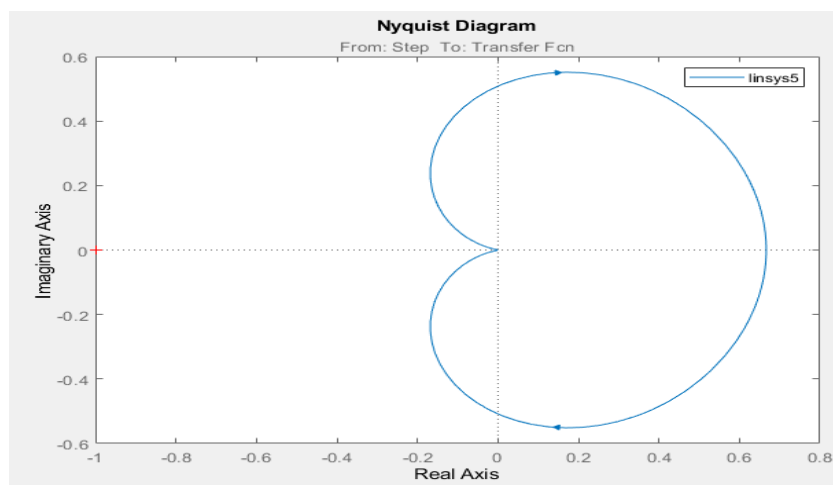


Рисунок 14.6- Диаграмма Найквиста

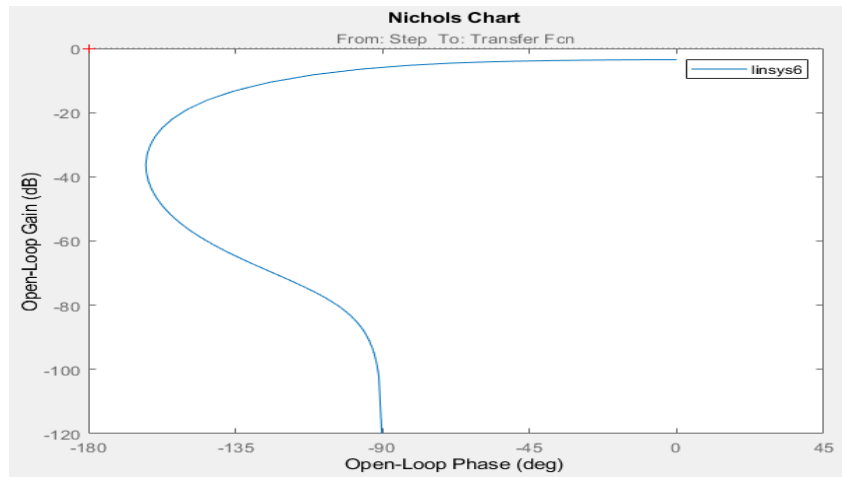


Рисунок 14.7- Частотная характеристика Николса

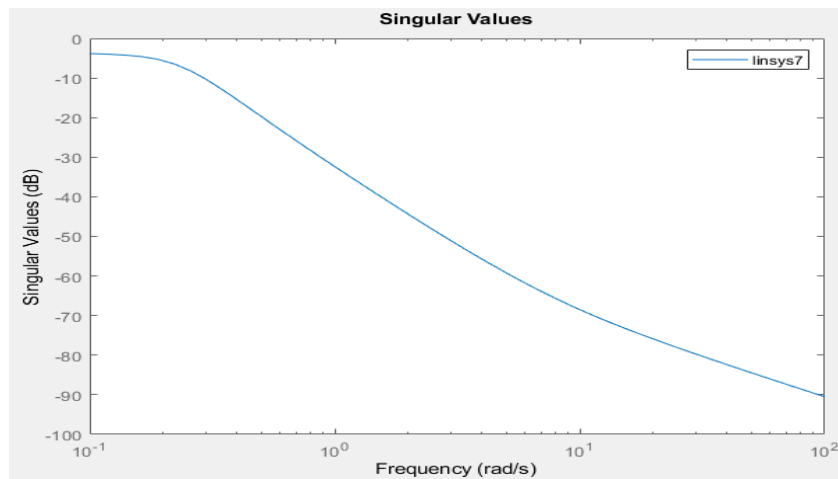


Рисунок 14.8- Сингулярные числа

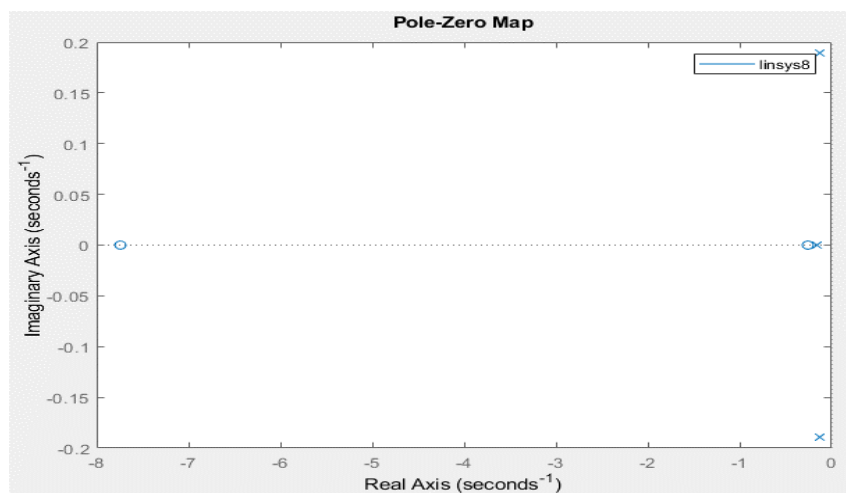


Рисунок 14.9- Диаграмма распределения полюсов-нулей передаточной функции модели

Литература

Литература к лекции №1

1. Ганкин М.З. Комплексная автоматизация и АСУТП водохозяйственных систем. - М.: Агропромиздат, 1991. -432 с.
2. Блок-схемы алгоритмов. ГОСТ. Примеры. URL: <https://pro-prof.com/archives/1462?ysclid=lbtxh1eas4886725237>.
- 3.ГОСТ 19.701–90 (ИСО 5807–85) «Единая система программной документации».
- 4.Алгоритм. Свойства алгоритма. URL: <https://pro-prof.com/archives/578>
- 5.Алгоритмы сортировки слиянием и быстрой сортировки. URL: <https://pro-prof.com/archives/813>.
- 6.yEd Graph Editor. URL: <https://www.yworks.com/products/yed>
- 7.Книги: алгоритмы. URL: <https://pro-prof.com/books-algorithms>
- 8.Рамбо Дж., Якобсон А., Буч Г. UML: специальный справочник. -СПб.: Питер, 2002. -656 с.
- 9.Шилов Н.В. Верификация шаблонов алгоритмов для метода отката и метода ветвей и границ. Моделирование и анализ информационных систем, ISSN 1818 - 1015, т.18, №4, 2011.
- 10.Брукс Ф., Мифический человек - месяц или как создаются программные системы. СПб. Символ Плюс, 1999 - 304 с. ил.

Литература к лекции №2

11. Гаспарян О.Н. MATLAB. Учебное пособие. Государственный инженерный университет Армении. 2005.- 143 с.
- 12.Общие сведения о matlab. Пакеты расширения matlab. URL: <https://studfile.net/preview/730984/page:2/>.
13. Русак Л.В., Снисаренко С.В., Стасевич Н.А. Основы математического моделирования в пакете MATLAB. Минск. БГУИР. 2010- 117 с.
- 14.Кетков Ю. Л., Кетков А. Ю., Шульц М. М. MATLAB 7: программирование, численные методы. - СПб.: БХВ-Петербург, 2005. 752 с.

Литература к лекциям №8

1. Ануфриев И.Е., Смирнов А.Б., Смирнова Е.Н. MATLAB 7 / И.Е. Ануфриев, А.Б. Смирнов, Е.Н. Смирнова. – СПб.: БХВ-Петербург, 2005. - 1104 с.
2. Дьяконов В. П. MATLAB. Полный самоучитель.- М.: ДМК Пресс, 2012. - 768 с.
3. Дьяконов В. П. Simulink 5/6/7: Самоучитель.- М.: ДМК-Пресс, 2008. - 784 с.
4. Золотых Н.Ю. Использование пакета Matlab в научной и учебной работе. Учебно-методические материалы по программе повышения квалификации «Информационные технологии и компьютерная математика». Нижний Новгород, 2006.- 165 с.
5. Малахов А. А. Основы работы с вычислительной системой Matlab и пакетом визуального моделирования Simulink. Учебное пособие - [электронный курс]. «Московский государственный технический университет имени Н.Э. Баумана» (МГТУ им. Н.Э. Баумана), 2017.- 52 с. <https://baumanka.pashinin.com>
6. Наместников С.М. Основы программирования в MatLab / Сборник лекций: УлГТУ, Ульяновск, 2011.- 39 с. https://scask.ru/a_lect_matlab.php

Литература к лекциям №9-11

1. Ануфриев И.Е., Смирнов А.Б., Смирнова Е.Н. MATLAB 7 / И.Е. Ануфриев, А.Б. Смирнов, Е.Н. Смирнова. - СПб.: БХВ-Петербург, 2005. - 1104 с.
2. Дьяконов В.П. MATLAB и Simulink для радиоинженеров. - М.: ДМК-Прогресс, 2011. - 976 с.
3. Малахов А. А. Основы работы с вычислительной системой Matlab и пакетом визуального моделирования Simulink. Учебное пособие - [электронный курс]. «Московски О. Н. й государственный технический университет имени Н.Э. Баумана» (МГТУ им. Н.Э. Баумана), 2017.- 52 с. <https://baumanka.pashinin.com>
4. Русак Л.В. Основы математического моделирования в пакете MATLAB: метод. Пособие к практ. занятиям по дисц. «Учебная практика» для студ. спец. «Информационные технологии и управление в технических

- системах» днев. Формы обуч. / Л.В. Русак, С.В. Снисаренко, Н.А. Стасевич. - Минск: БГУИР, 2010.- 64 с.
5. Васильев В.В., Симак Л.А., Рыбникова А.М. Математическое и компьютерное моделирование процессов и систем в среде MATLAB/SIMULINK. Учебное пособие для студентов и аспирантов / В.В. Васильев, Л.А. Симак, А.М. Рыбникова. - К.: НАН Украины, 2008. - 91 с.
 6. Щербаков В.С., Руппель А.А., Глушец В.А. Основы моделирования систем автоматического регулирования и электротехнических систем в среде MATLAB и SIMULINK: Учебное пособие. - Омск: Изд-во СибАДИ, 2003.- 160 с.
 7. Шульгин В.И. Основы Работы с SIMULINK MATLAB Моделирование на ЭВМ и использованием пакета MATLAB/ Лабораторный практикум. Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Харьков. 2010. - 60 с.
 8. Герасимов А. И., Регеда В. В., Регеда О. Н. Моделирование в среде MATLAB-Simulink : метод. указания к лабораторным работам / сост.: А. И. Герасимов, В. В. Регеда, О. Н. Регеда. - Пенза : Изд-во ПГУ, 2017. - 104 с.

Варианты для самостоятельного выполнения

Варианты к практическому занятию №1

№ варианта	№ пункта задания	
	1	2
1	$\frac{ctg 3,5 + \sqrt{e^{3,6} + \sin 8,5\pi}}{(\cos 1,5\pi - 3ctg 3,6)^2} + e^{-6} \cdot (5 + \sin 3,6\pi)$	$\frac{\frac{\sin 9,95\pi}{\ln 3,6} + \frac{\sqrt{\ln 3,6}}{\sin 5,4\pi} - \sqrt{\frac{ ctg 5,4 }{\ln 3,6}}}{\sqrt{\frac{ tg 5,8 }{\ln 3,6}} - \left(\frac{\sin 4,6\pi}{\lg 3,4}\right)}$
2	$\frac{\sqrt{5ctg 4,5} - \sin 3,5\pi}{\cos 0,6\pi + (\lg 15,6)^2} - \sqrt{\lg 25,6} \cdot (1 + e^{-5})$	$\frac{\left(\frac{\sin 5,95\pi}{\cos 3,6\pi}\right)^2 - \frac{ctg 7,45}{\sin 5,4\pi} + \frac{\ln 3,6}{4 \ln 7,6}}{\frac{\sin 4,\pi}{\cos 3,6\pi} + \sqrt{\frac{ctg 7,46}{ \sin 5,4\pi }}}$
3	$\frac{tg 3,5 + \sqrt{5 + \sin 2,5}}{(\sin 2,5\pi - 4ctg 3,6)^4} + e^{-6/15} \cdot (2 + \sin 3,8\pi)$	$\frac{\frac{\sqrt{\ln 3,5}}{\sin 6,5} + \sqrt{\frac{ \sin 10,5 }{\ln 3,6}}}{\sqrt{\frac{\ln 5,8}{\ln 4,6}}}$
4	$\frac{\sqrt{ tg 8,5 } - \cos 1,5\pi}{1,5\pi + (\ln 36)^2} - \sqrt{\ln 18,4} \cdot (5 - e^{2/5})$	$\frac{\sqrt{\frac{\ln 3,6}{2 \lg 5,6 }} + \left(\frac{\sin 4,5\pi}{\cos 5,6\pi} - \frac{ctg 3,56}{\cos 5,6}\right)}{\left(\frac{ctg 5,8}{\cos 5,6\pi}\right)^2}$
5	$\frac{ctg 3,5 + \sqrt{e^{3,6} + \sin 8,5\pi}}{(\cos 1,5\pi - 3ctg 3,6)^2} + e^5 \cdot (5 + \sin 3,6\pi)$	$\frac{\frac{\cos 9,95\pi}{\ln 3,6} + \frac{\sqrt{ \lg 3,6 }}{\sin 5,6\pi} - \sqrt{\frac{ tg 5,4 }{\ln 4,6}}}{\sqrt{\frac{ tg 5,8 }{\ln 3,6}} - \left(\frac{\cos 4,6\pi}{\lg 3,4}\right)^{3,6}}$
6	$\frac{\sqrt{ \sin 4,8 } - \sin 1,5\pi}{0,905\pi + (\ln 15)^2} - \sqrt{\ln 46} \cdot (5 - e^{6/15})$	$\frac{\sqrt{\frac{ \sin 5,6 }{4 \ln 9,6}} + \left(\frac{\sin 4,6\pi}{\cos 3,6\pi} - \frac{ctg 3,4}{\sin 5,6}\right)}{\left(\frac{ctg 5,6}{\sin 5,4\pi}\right)}$
7	$\frac{\sin 7,2 + \sqrt{6 + \cos 1,5}}{(\sin 1,5\pi - 3ctg 4,6)^3} + e^{-1/5} \cdot (6 + \sin 5\pi)$	$\frac{\frac{\sqrt{\ln 5,6}}{\cos 5,4} + \sqrt{\frac{ \sin 10,45 }{\ln 3,6}}}{\sqrt{\frac{\ln 5,6}{\ln 10,46}}}$
8	$\frac{\sqrt{ \sin 8,5 } - \cos 1,5\pi}{1,9\pi + (\ln 32)^2} - \sqrt{\ln 18,6} \cdot (6 - e^{-1/5})$	$\frac{\sqrt{\frac{ \sin 3,2 }{5 \ln 3,6}} + \left(\frac{\sin 8,6\rho}{\cos 5,4\pi} - \frac{tg 4,6}{\sin 10,6}\right)}{\left(\frac{ctg 7,6}{\sin 8,6\pi}\right)^4}$

9	$\frac{\sqrt{10 ctg8,5 } - \cos 3,4\pi}{\sin 1,5\pi + (\ln 14,4)^2} - \sqrt{\ln 18,6} \cdot (1,5 - e^{-1/5})$	$\frac{\left(\frac{\sin 4,6\pi}{\cos 8,5\pi}\right)^2 - \frac{ctg 4,46}{\sin 5,4\pi} + \frac{\ln 3,4}{6 \ln 4,6}}{\frac{\sin 4,2\pi}{\cos 4,6\pi} + \sqrt{\frac{ctg 4,6}{\cos 3,4\pi}}}$
10	$\frac{ctg 3,6 + \sqrt{2 + \sin 1,5}}{(\cos 1,5\pi - 3ctg 3,6)^2} - e^{-1/5} \cdot (6 + \cos 3,6\pi)$	$\frac{\frac{\sqrt{ \lg 3,6 }}{\sin 5,4} + \frac{\sqrt{ \cos 5,6 }}{\sqrt{ \lg 3,6 }}}{\sqrt{\frac{\lg 5,4}{\lg 3,6}}}$
11	$\frac{tg 4,6 + \sqrt{e^{5,6} + \sin 4,6\pi}}{(\sin 1,5\pi - 5ctg 3,6)^2} - e^4 \cdot (5 + \sin 3,6\pi)$	$\frac{\frac{\cos 4,6\pi}{\lg 3,6} + \frac{\sqrt{\lg 15}}{\cos 5,4\pi} - \frac{\sqrt{ \lg 5,6 }}{\sqrt{ \lg 4,6 }}}{\sqrt{\frac{ \lg 5,6 }{ \lg 3,6 }} - \left(\frac{\cos 9,96\pi}{\lg 3,4}\right)^4}$
12	$\frac{\sqrt{ \cos 3,6 } - \sin 8,5\pi}{3,6\pi + (\ln 36)^2} - \sqrt{\ln 4,6} \cdot (6 - e^{1/5})$	$\frac{\sqrt{\frac{ \sin 4,2 }{5 \ln 3,6}} + \left(\frac{\sin 4,2\pi}{\cos 5,4\pi} - \frac{ctg 5,6}{\sin 3,6}\right)}{\left(\frac{ctg 4,6}{\sin 5,6\pi}\right)^2}$
13	$\frac{tg 3,5 + \sqrt{e^{3,6} + \sin 8,5\pi}}{(\sin 3,6\pi - 3tg 1,8)^2} - e^{2/3} \cdot (4 + \sin 3,6\pi)$	$\frac{\frac{\cos 9,95\pi}{\ln 3,6} + \frac{\sqrt{\ln 3,6}}{\cos 5,4\pi} - \frac{\sqrt{ \lg 4,8 }}{\sqrt{\ln 3,6}}}{\sqrt{\frac{ \lg 5,6 }{\ln 3,6}} - \left(\frac{\sin 9,96\pi}{\lg 4,6}\right)^4}$
14	$\frac{\sqrt{6 tg 3,6 } - \cos 1,5\pi}{\sin 4,6\pi + (\ln 36)^2} - \sqrt{\ln 15} \cdot (1 - e^{1/2})$	$\frac{\left(\frac{\sin 5,6\pi}{\cos 3,6\pi}\right)^4 - \frac{ctg 3,6}{\sin 5,4\pi} + \frac{\lg 3,6}{5 \lg 15,46}}{\frac{\sin 5,6\pi}{\cos 3,6\pi} + \sqrt{\frac{ \lg 4,6 }{ \sin 5,4\pi }}}$
15	$\frac{tg 3,6 + \sqrt{5 + \sin 4,6}}{(\sin 1,5\pi - 5tg 3,6)^4} - e^{1/2} \cdot (3,6 + \sin 4,6\pi)$	$\frac{\frac{\sqrt{ \lg 3,6 }}{\sin 9,96} + \frac{\sqrt{ \cos 5,6 }}{\sqrt{ \lg 4,6 }}}{\sqrt{\frac{\lg 5,6}{ \lg 3,6 }}}$

Варианты к практическому занятию №2

№ варианта	№ пункта задания				
	1	2-3	4	5	6
1	$a = \begin{pmatrix} 9,4 \\ 8,9 \\ 1,5 \end{pmatrix}; b = \begin{pmatrix} 7,4 \\ 2,6 \\ 4,5 \end{pmatrix}$	$a = (0,5 \ 8,4 \ 5,6 \ 3,4 \ 6,5 \ 1,5 \ 0,8 \ 10 \ 15);$ $N=4,6$	$a = \begin{pmatrix} 3,6 \\ 2,5 \\ 1,8 \\ 0,6 \\ 9,8 \\ 5,4 \end{pmatrix}$	$a = (-0,5 \ 4,5 \ -9,6 \ 3,8 \ 7,6 \ 4)$	$a = (4 \ 8 \ -5 \ 6);$ $b = (-4 \ 6 \ -8 \ 5);$ $e = (6 \ 8 \ 4 \ 5);$ $N=3,6; L=4$
2	$a = \begin{pmatrix} 5 \\ 6,4 \\ 9,5 \end{pmatrix}; b = \begin{pmatrix} 4,4 \\ 0,6 \\ 8,6 \end{pmatrix}$	$a = (6,4 \ 3,6 \ 9,9 \ 2,4 \ 6,5 \ 6 \ 3,8 \ 1,4 \ 0,2);$ $N=4,2$	$a = \begin{pmatrix} 1,8 \\ 7,2 \\ 3,6 \\ 8,4 \\ 7,4 \\ 5,8 \end{pmatrix}$	$a = (1,8 \ -9,6 \ 2,6 \ 9,4 \ -2,4 \ 3,8)$	$a = (0,8 \ 6,4 \ -1,8 \ 0,4);$ $b = (7 \ 3 \ -6 \ 4);$ $e = (3,8 \ 6 \ 0,2 \ 8);$ $N=7,4; L=8$
3	$a = \begin{pmatrix} 8,2 \\ 9 \\ 1,2 \end{pmatrix}; b = \begin{pmatrix} 9,2 \\ 6 \\ 0,4 \end{pmatrix}$	$a = (2,6 \ 5,4 \ 9,6 \ 9,8 \ 1,5 \ 9 \ 9,4 \ 4,8 \ 8);$ $N=1,5$	$a = \begin{pmatrix} 4,2 \\ 9,2 \\ 7,9 \\ 9,6 \\ 6,5 \\ 0,4 \end{pmatrix}$	$a = (8,5 \ 9,4 \ 6,4 \ 7,6 \ 7,4 \ 3,8)$	$a = (6,5 \ 1,8 \ -7 \ 0,4);$ $b = (6 \ 3,2 \ -9,4 \ 0,4);$ $e = (2 \ 4 \ 7 \ 8);$ $N=4,6; L=6$
4	$a = \begin{pmatrix} 8,2 \\ 9 \\ 1,4 \end{pmatrix}; b = \begin{pmatrix} 9,4 \\ 6 \\ 0,8 \end{pmatrix}$	$a = (3,6 \ 4,6 \ 8,2 \ 8,6 \ 1,5 \ 9,4 \ 9,6 \ 4,2 \ 8);$ $N=4,5$	$a = \begin{pmatrix} 4,2 \\ 8 \\ 7,8 \\ 9,6 \\ 6,4 \\ 0,4 \end{pmatrix}$	$a = (8,4 \ 9,4 \ 6 \ 7,6 \ 7,4 \ 3,8)$	$a = (6,4 \ 1,8 \ -7 \ 0,4);$ $b = (2,6 \ 0,4 \ -0,9 \ 8);$ $e = (6 \ 3 \ 9 \ 4);$ $N=3,8; L=4$

5	$a = \begin{pmatrix} 8,1 \\ 9 \\ 1,2 \end{pmatrix}; b = \begin{pmatrix} 9,1 \\ 6,3 \\ 0,9 \end{pmatrix}$	$a = (2,7 \ 3,4 \ 7,5 \ 9,6 \ 2,5 \ 5,7 \ 9,5 \ 3,8 \ 5);$ $N=1,4$	$a = \begin{pmatrix} 4,2 \\ 9,1 \\ 7,9 \\ 9,5 \\ 6,5 \\ 0,4 \end{pmatrix}$	$a = (8,4 \ 9,3 \ 6,7 \ 7,5 \ 3,4 \ 3,9)$	$a = (6,5 \ 1,7 \ -2 \ 0,3);$ $b = (2,7 \ 0,4 \ -0,4 \ 3,2);$ $e = (6 \ 3,2 \ 9,5 \ 0,4);$ $N=4,3; L=3,8$
6	$a = \begin{pmatrix} 8,4 \\ 6 \\ 1,6 \end{pmatrix}; b = \begin{pmatrix} 7,2 \\ 5,4 \\ 1,8 \end{pmatrix}$	$a = (1,5 \ 2,6 \ 5 \ 3,4 \ 8,6 \ 5,4 \ 8,2 \ 9 \ 0,6);$ $N=5$	$a = \begin{pmatrix} 3,6 \\ 8,2 \\ 1,5 \\ 7,6 \\ 9 \\ 0,8 \end{pmatrix}$	$a = (5,4 \ 2,6 \ 3,2 \ 9 \ 7,6 \ 8)$	$a = (5,4 \ -2 \ 4,6 \ 8);$ $b = (3,6 \ 5 \ -4 \ 1,8);$ $e = (3,2 \ 2,6 \ 5 \ 8 \ 4);$ $N=1,5; L=4$
7	$a = \begin{pmatrix} 5,4 \\ 2 \\ 1,8 \end{pmatrix}; b = \begin{pmatrix} 2,6 \\ 6 \\ 1,5 \end{pmatrix}$	$a = (0,6 \ 7,4 \ 4,6 \ 2,4 \ 6,4 \ 1,4 \ 4,8 \ 9 \ 4);$ $N=3,6$	$a = \begin{pmatrix} 7,6 \\ 4,5 \\ 1,8 \\ 3,6 \\ 5,8 \\ 1,4 \end{pmatrix}$	$a = (3,4 \ 5,6 \ 4,2 \ 6 \ 7,4 \ 8)$	$a = (5 \ 8 \ -4 \ 6);$ $b = (3,6 \ 5 \ -6 \ 2);$ $e = (4 \ 5 \ 8 \ 6);$ $N=6,4; L=5$
8	$a = \begin{pmatrix} 1,4 \\ 9 \\ 0,8 \end{pmatrix}; b = \begin{pmatrix} 8,4 \\ 2 \\ 7,6 \end{pmatrix}$	$a = (7,4 \ 5,6 \ 7,2 \ 2,4 \ 6,5 \ 6 \ 3,8 \ 1,4 \ 0,2);$ $N=1,8$	$a = \begin{pmatrix} 4,8 \\ 7,2 \\ 3,6 \\ 8,4 \\ 7,4 \\ 5,6 \end{pmatrix}$	$a = (0,6 \ 4,5 \ -9,6 \ 3,8 \ 7,6 \ 0,4)$	$a = (0,6 \ 3,4 \ -1,8 \ 2,6);$ $b = (4 \ 1,5 \ -6 \ 2);$ $e = (3 \ 5 \ 8 \ 4);$ $N=4,2; L=2$

9	$a = \begin{pmatrix} 9,4 \\ 8,9 \\ 1,5 \end{pmatrix}; b = \begin{pmatrix} 8,4 \\ 2 \\ 7,6 \end{pmatrix}$	$a = (3,4 \ 4,6 \ 8,6 \ 9,8 \ 1,5 \ 6 \ 7,4 \ 1,8 \ 2);$ $N=6$	$a = \begin{pmatrix} 3,2 \\ 5,4 \\ 7,9 \\ 9,6 \\ 6,5 \\ 0,8 \end{pmatrix}$	$a = (3,8 \ 9, \ 2,6 \ 9,4 \ -2,4 \ 3,8)$	$a = (4,5 \ 3,8 \ -5 \ 0,6);$ $b = (0,4 \ 6 \ -8 \ 5);$ $e = (4 \ 2 \ 5 \ 6);$ $N=7,4; L=4$
10	$a = \begin{pmatrix} 5 \\ 6,4 \\ 9,5 \end{pmatrix}; b = \begin{pmatrix} 2,6 \\ 6 \\ 1,5 \end{pmatrix}$	$a = (2,6 \ 7,2 \ 8,2 \ 8,6 \ 1,5 \ 9,4 \ 9,6 \ 4,2 \ 6);$ $N=8,2$	$a = \begin{pmatrix} 4,2 \\ 6 \\ 7,6 \\ 4,6 \\ 6,4 \\ 0,6 \end{pmatrix}$	$a = (1,5 \ 9,6 \ 2,6 \ 5,4 \ -4,6 \ 3,8)$	$a = (4,6 \ 1,5 \ -8 \ 0,4);$ $a = (8 \ 6,4 \ -1,8 \ 0,4);$ $e = (1,5 \ 9 \ 4 \ 6);$ $N=5,4; L=3,6$
11	$a = \begin{pmatrix} 8,2 \\ 9 \\ 1,2 \end{pmatrix}; b = \begin{pmatrix} 7,2 \\ 5,4 \\ 1,8 \end{pmatrix}$	$a = (7,6 \ 4,6 \ 5 \ 8,6 \ 4,5 \ 9,4 \ 5,6 \ 4,2 \ 1,8);$ $N=5,4$	$a = \begin{pmatrix} 3,2 \\ 9,1 \\ 7,9 \\ 4,5 \\ 6,5 \\ 0,8 \end{pmatrix}$	$a = (3,6 \ 8,4 \ 6,4 \ 7,6 \ 7,4 \ 4,8)$	$a = (4,5 \ 1,7 \ -2 \ 0,3);$ $b = (6 \ 5,8 \ -9,4 \ 0,4);$ $e = (4,8 \ 6 \ 0,2 \ 8);$ $N=4,2; L=1,5$
12	$a = \begin{pmatrix} 8,2 \\ 9 \\ 1,4 \end{pmatrix}; b = \begin{pmatrix} 9,1 \\ 6,3 \\ 0,9 \end{pmatrix}$	$a = (2,6 \ 5,4 \ 7,5 \ 9,6 \ 4,5 \ 5 \ 6,5 \ 3,8 \ 6);$ $N=3,6$	$a = \begin{pmatrix} 4,6 \\ 7,2 \\ 1,5 \\ 7,6 \\ 9 \\ 0,6 \end{pmatrix}$	$a = (3,6 \ 1,4 \ 5 \ 3,6 \ 7,4 \ 5,8)$	$a = (5,4 \ -2 \ 3,6 \ 8);$ $b = (3,6 \ 4 \ -0,9 \ 8,2);$ $e = (5 \ 3 \ 9 \ 4);$ $N=2,6; L=3,4$

13	$a = \begin{pmatrix} 8,1 \\ 9 \\ 1,2 \end{pmatrix}; b = \begin{pmatrix} 7,4 \\ 2,6 \\ 4,5 \end{pmatrix}$	$a = (4,5 \ 2,6 \ 6 \ 3,4 \ 8,6 \ 5,4 \ 9,2 \ 9 \ 3,6);$ $N=1,5$	$a = \begin{pmatrix} 4,6 \\ 7,5 \\ 4,8 \\ 3,2 \\ 5,4 \\ 1,5 \end{pmatrix}$	$a = (5,4 \ 8,6 \ 1,5 \ 7,6 \ 3,4 \ 3,6)$	$a = (5 \ 3,6 \ -4 \ 6);$ $b = (5,6 \ 0,4 \ -0,4 \ 3,2);$ $e = (5 \ 3,2 \ 9,5 \ 0,4);$ $N=9; L=6$
14	$a = \begin{pmatrix} 8,4 \\ 6 \\ 1,6 \end{pmatrix}; b = \begin{pmatrix} 4,4 \\ 0,6 \\ 8,6 \end{pmatrix}$	$a = (4,5 \ 2,6 \ 5 \ 3,4 \ 8,6 \ 5,4 \ 8,2 \ 9 \ 1,8);$ $N=3,4$	$a = \begin{pmatrix} 3,8 \\ 4,2 \\ 5,6 \\ 8,4 \\ 7,4 \\ 8,6 \end{pmatrix}$	$a = (8,4 \ 2,6 \ 3,4 \ 9 \ 7,6 \ 8)$	$a = (0,4 \ 3,4 \ -1,8 \ 2,6);$ $b = (4,6 \ 5 \ -4 \ 1,8);$ $e = (3,8 \ 2,6 \ 5 \ 1,8 \ 4);$ $N=1,8; L=4,6$
15	$a = \begin{pmatrix} 5,4 \\ 2 \\ 1,8 \end{pmatrix}; b = \begin{pmatrix} 9,2 \\ 6 \\ 0,4 \end{pmatrix}$	$a = (5,4 \ 5,6 \ 3,2 \ 2,4 \ 4,5 \ 6 \ 3,8 \ 1,4 \ 4,2);$ $N=4,6$	$a = \begin{pmatrix} 4,2 \\ 3,4 \\ 7,9 \\ 9,6 \\ 4,5 \\ 0,6 \end{pmatrix}$	$a = (0,8 \ 4,5 \ -3,6 \ 4,8 \ 7,6 \ 0,6)$	$a = (4,5 \ 3,8 \ -5 \ 0,6);$ $b = (4 \ 1,8 \ -6 \ 2);$ $e = (4,8 \ 2 \ 5 \ 6);$ $N=3,4; L=5.6$

Варианты к практическому занятию №3

№ варианта	№ пункта задания					
	1	2	3	4	5-7	8
1	$y(x) = \frac{tg 5,6x + \sqrt{6 + \sin 7,6x}}{(\sin 4,5x - 3,5tg 13,6x)^2} - e^{-1,5x};$ $x = 0,1 \ 0,4 \ 0,6 \ 1,2 \ 1,5 \ 3,6$	$h = 0,2;$ $h2 = 0,4;$ $a = 2$	$a = -1;$ $b = 1;$ $h = 0,01$	$y(x) = \sin \frac{0,5}{2x^2};$ $z(x) = \cos \frac{2,5}{2x^4};$ $x \in [-1, 1]$	$a = \begin{bmatrix} 4,8 \\ 4,5 \\ 0,8 \end{bmatrix}; \quad b = \begin{bmatrix} 5,6 \\ -3,6 \\ 1,8 \end{bmatrix}$	$a = [7,6;-3,5;1,5];$ $b = [-2,4;1,5;7,8];$ $c = [-0,4;-4,2;5,4]$
2	$y(x) = \frac{ctg 3,6x + \sqrt{5 + \cos 4,6x}}{(\cos 1,5x - 5ctg 3,6x)^4} - e^{-2x};$ $x = 0,2 \ 0,6 \ 1,8 \ 3,2 \ 4,5 \ 5,8$	$h = 0,1;$ $h2 = 0,2;$ $a = 4$	$a = -2;$ $b = 1,5;$ $h = 0,02;$	$y(x) = \cos \frac{1,5}{3x^2};$ $z(x) = \sin \frac{3,5}{4x^2};$ $x \in [-2, 2]$	$a = \begin{bmatrix} 1,8 \\ 5 \\ 0,4 \end{bmatrix}; \quad b = \begin{bmatrix} 3,4 \\ -1,6 \\ 3,5 \end{bmatrix}$	$a = [3,6;-5,4;3,6];$ $b = [-3,4;3,6;8,6];$ $c = [-3,6;-7,5;8,6]$
3	$y(x) = \frac{ctg 3,6x + \sqrt{4,5 + \sin 3,6x}}{(\sin 2,5x - 5tg 3,6x)^2} - e^{-3,6x};$ $x = 0,2 \ 0,4 \ 1,8 \ 3,2 \ 4,6 \ 4,8$	$h = 0,1;$ $h2 = 0,4;$ $a = 6$	$a = 1$ $b = 3,6$ $h = 0,02$	$y(x) = tg \frac{1,5}{2x^2};$ $z(x) = ctg \frac{3,5}{4x^2};$ $x \in [-1, 1]$	$a = \begin{bmatrix} 3,2 \\ 4,5 \\ 1,8 \end{bmatrix}; \quad b = \begin{bmatrix} 3,4 \\ -3,6 \\ 4,5 \end{bmatrix}$	$a = [12,6;-8,5;5,6];$ $b = [-3,4;2,6;7,6];$ $c = [-0,4;-1,5;5,6]$

4	$y(x) = \frac{\cos 3,6x + \sqrt{5 + \cos 4,6x}}{(tg 1,5x - 5ctg 3,6x)^3} - e^{-1,8x};$ $x = 0,4 \ 0,9 \ 1,8 \ 3,2 \ 4,5 \ 5,8$	$h = 0,2;$ $h2 = 0,2;$ $a = 4$	$a = 2$ $b = 4$ $h = 0,05$	$y(x) = \sin \frac{4,5}{2x^4};$ $z(x) = \cos \frac{1,8}{5x^3};$ $x \in [-3, 3,5]$	$a = \begin{bmatrix} 3,8 \\ 2,4 \\ 0,8 \end{bmatrix}; b = \begin{bmatrix} 5,6 \\ -3,6 \\ 1,8 \end{bmatrix}$	$a = [5,6;-1,5;0,6];$ $b = [-3,4;3,6;5,6];$ $c = [-0,6;-1,5;4,6]$
5	$y(x) = \frac{\sin 3,6x + \sqrt{3,5 + \cos 3,6x}}{(ctg 4,5x - 3tg 7,6x)^4} - e^{-3,5x};$ $x = 0,2 \ 0,6 \ 0,8 \ 1,4 \ 3,5 \ 4,8$	$h = 0,4;$ $h2 = 0,1;$ $a = 2$	$a = 1$ $b = 3,6$ $h = 0,01$	$y(x) = \ln \frac{2,5}{2x^2};$ $z(x) = \cos \frac{1,5}{1,4x^2};$ $x \in [1, 5]$	$a = \begin{bmatrix} 4,5 \\ 3,6 \\ 2,6 \end{bmatrix}; b = \begin{bmatrix} 3,4 \\ -0,6 \\ 1,5 \end{bmatrix}$	$a = [4,6;-0,5;3,6];$ $b = [-1,8;2,6;5,4];$ $c = [-0,6;-3,5;4,6]$
6	$y(x) = \frac{\cos 4,5x + \sqrt{2,6 + \sin 1,5x}}{(ctg 1,5x - 3,6tg 4,6x)^4} - e^{-0,5x};$ $x = 0,1 \ 0,4 \ 0,6 \ 1,2 \ 1,4 \ 3,8$	$h = 0,4;$ $h2 = 0,5;$ $a = 6$	$a = 3,4$ $b = 4,6$ $h = 0,02$	$y(x) = \ln \frac{0,5}{2x^2};$ $z(x) = \sin \frac{2,5}{3x^2};$ $x \in [1, 4]$	$a = \begin{bmatrix} 2,6 \\ 1,4 \\ 3,6 \end{bmatrix}; b = \begin{bmatrix} 5,4 \\ -0,6 \\ 3,6 \end{bmatrix}$	$a = [7,6;-5,6;3,6];$ $b = [-2,4;2,6;7,6];$ $c = [-1,4;-3,5;5,6]$
7	$y(x) = \frac{ctg 2,6x + \sqrt{5,4 + \cos 3,6x}}{(\sin 2,6x - 4ctg 3,6x)^4} - e^{-0,36x};$ $x = 0,2 \ 0,4 \ 0,6 \ 0,8 \ 1,2 \ 1,4$	$h = 0,2;$ $h2 = 0,1;$ $a = 4$	$a = 1,4$ $b = 3,6$ $h = 0,01$	$y(x) = \sin \frac{1}{3x^2};$ $z(x) = \lg \frac{2,5}{4x^2};$ $x \in [2, 4]$	$a = \begin{bmatrix} 2,6 \\ 1,8 \\ 4,5 \end{bmatrix}; b = \begin{bmatrix} 5,6 \\ -3,6 \\ 4,5 \end{bmatrix}$	$a = [4,6;-1,5;2,6];$ $b = [-3,4;0,6;5,4];$ $c = [-0,6;-3,5;5,6]$

8	$y(x) = \frac{tg 3,6x + \sqrt{5 + \cos 4,6x}}{(\cos 1,5x - 5tg 3,6x)^4} - e^{-1,5x};$ $x = 0,4 \ 0,8 \ 1,2 \ 1,5 \ 1,8 \ 3,6$	$h = 0,1;$ $h2 = 0,2;$ $a = 6$	$a = 1,5$ $b = 4,6$ $h = 0,02$	$y(x) = \lg \frac{0,5}{2x^2};$ $z(x) = tg \frac{3,5}{4x^3};$ $x \in [4, 6]$	$a = \begin{bmatrix} 1,5 \\ 4,6 \\ 3,6 \end{bmatrix}; b = \begin{bmatrix} 3,6 \\ -0,6 \\ 1,5 \end{bmatrix}$	$a = [3,6; -4,5; 7,6];$ $b = [-3,4; 2,6; 5,6];$ $c = [-0,4; -1,5; 4,6]$
9	$y(x) = \frac{\sin 4,6x + \sqrt{3,5 + \cos 4,6x}}{(tg 1,5x - 3,5 \cos 4,6x)^4} - e^{-1,8x};$ $x = 0,1 \ 0,5 \ 1 \ 1,5 \ 2 \ 2,5$	$h = 0,4;$ $h2 = 0,2;$ $a = 2$	$a = -2$ $b = 2$ $h = 0,02$	$y(x) = \ln \frac{1}{3x^3};$ $z(x) = ctg \frac{2,5}{4x^2};$ $x \in [5, 10]$	$a = \begin{bmatrix} 1,8 \\ 5,6 \\ 0,4 \end{bmatrix}; b = \begin{bmatrix} 4,5 \\ -0,6 \\ 5,4 \end{bmatrix}$	$a = [4,6; -0,5; 3,6];$ $b = [-1,5; 0,6; 5,6];$ $c = [-5,6; -4,5; 1,8]$
10	$y(x) = \frac{ctg 7,6x + \sqrt{1,5 + \sin 3,6x}}{(\cos 3,5x - 4,5ctg 4,6x)^3} - e^{-3,6x};$ $x = 0,2 \ 0,4 \ 0,6 \ 0,8 \ 1 \ 1,2$	$h = 0,1;$ $h2 = 0,1;$ $a = 6$	$a = 1$ $b = 4,5$ $h = 0,01$	$y(x) = \lg \frac{3}{4x^2};$ $z(x) = \sin \frac{1,5}{4x^3};$ $x \in [2, 6]$	$a = \begin{bmatrix} 1,5 \\ 4,6 \\ 0,6 \end{bmatrix}; b = \begin{bmatrix} 3,6 \\ -0,6 \\ 1,8 \end{bmatrix}$	$a = [3,6; -1,5; 2,6];$ $b = [-1,4; 2,6; 5,4];$ $c = [-0,4; -3,5; 4,6]$
11	$y(x) = \frac{ctg 5,6x + \sqrt{1,5 + \cos 3,6x}}{(\sin 4,5x - 7,5ctg 4,6x)^4} - e^{-1,5x};$ $x = 0,2 \ 0,6 \ 1 \ 1,4 \ 1,8 \ 2,4$	$h = 0,2;$ $h2 = 0,4;$ $a = 2$	$a = 3$ $b = 4,6$ $h = 0,02$	$y(x) = ctg \frac{1}{2x^3};$ $z(x) = \sin \frac{1,5}{4x^4};$ $x \in [1, 4]$	$a = \begin{bmatrix} 5,4 \\ 4,5 \\ 1,5 \end{bmatrix}; b = \begin{bmatrix} 2,6 \\ -1,5 \\ 3,6 \end{bmatrix}$	$a = [4,6; -1,5; 2,6];$ $b = [-3,4; 0,5; 5,6];$ $c = [-0,6; -4,5; 3,6]$

12	$y(x) = \frac{\sin 3,6x + \sqrt{5 + \cos 4,6x}}{(\cos 1,5x - 5 \sin 3,6x)^4} - e^{-3,6x};$ $x = 0,1 \ 0,5 \ 0,9 \ 1,3 \ 1,8 \ 2,4$	$h = 0,1;$ $h_2 = 0,5;$ $a = 4$	$a = -1,5$ $b = 2,6$ $h = 0,01$	$y(x) = tg \frac{1,5}{2x^2};$ $z(x) = lg \frac{3,5}{4x^3};$ $x \in [1, 5]$	$a = \begin{bmatrix} 3,6 \\ 4,5 \\ 0,4 \end{bmatrix}; \quad b = \begin{bmatrix} 5,8 \\ -1,5 \\ 4,6 \end{bmatrix}$	$a = [3,4;-1,5;2,6];$ $b = [-3,4;0,6;4,8];$ $c = [-0,6;-1,5;5,6]$
13	$y(x) = \frac{ctg 3,6x + \sqrt{2,5 + \cos 4,6x}}{(\cos 1,5x - 3,5ctg 3,6x)^3} - e^{-1,5x};$ $x = 0,1 \ 0,5 \ 0,6 \ 1,2 \ 1,5 \ 1,8$	$h = 0,2;$ $h_2 = 0,2;$ $a = 2$	$a = 3,5$ $b = 4,6$ $h = 0,02$	$y(x) = \cos \frac{1}{3x^2};$ $z(x) = \sin \frac{2}{5x^2};$ $x \in [-5, 5]$	$a = \begin{bmatrix} 2,6 \\ 4 \\ 0,6 \end{bmatrix}; \quad b = \begin{bmatrix} 5,4 \\ -4,6 \\ 1,8 \end{bmatrix}$	$a = [5,4;-3,5;2,6];$ $b = [-1,5;0,6;5,4];$ $c = [-0,4;-4,8;3,6]$
14	$y(x) = \frac{\cos 1,5x + \sqrt{4 + \sin 4,6x}}{(\cos 4,5x - 4,5tg 1,5x)^2} - e^{-3,4x};$ $x = 0,1 \ 0,4 \ 0,8 \ 1,2 \ 1,5 \ 1,8$	$h = 0,4;$ $h_2 = 0,4;$ $a = 4$	$a = -2$ $b = 2$ $h = 0,01$	$y(x) = tg \frac{1}{2x^2};$ $z(x) = ctg \frac{1,5}{4x^2};$ $x \in [-2, 2]$	$a = \begin{bmatrix} 3,6 \\ 4,5 \\ 1,8 \end{bmatrix}; \quad b = \begin{bmatrix} 5,6 \\ -1,5 \\ 4,5 \end{bmatrix}$	$a = [1,8;-3,5;2,6];$ $b = [-1,5;2,6;5,4]$ $c = [-0,4;-3,6;5,4]$
15	$y(x) = \frac{ctg 4,6x + \sqrt{1,5 + \cos 3,6x}}{(\cos 4,5x - 6tg 4,5x)^4} - e^{-1,5x};$ $x = 0,1 \ 0,4 \ 0,6 \ 0,8 \ 1,2 \ 1,8$	$h = 0,2;$ $h_2 = 0,5;$ $a = 6$	$a = -2$ $b = 4$ $h = 0,01$	$y(x) = \ln \frac{1,5}{4x^2};$ $z(x) = \cos \frac{2,5}{6x^3};$ $x \in [2, 6]$	$a = \begin{bmatrix} 2,6 \\ 5,6 \\ 1,4 \end{bmatrix}; \quad b = \begin{bmatrix} 3,6 \\ -1,5 \\ 4,8 \end{bmatrix}$	$a = [8,6;-3,5;2,6];$ $b = [-1,5;0,4;3,6]$ $c = [-4,6;-1,5;7,6]$

Варианты к практическому занятию №4

№ варианта	№ пункта задания					
	1	2	3	4	5	6
1	$A = \begin{pmatrix} 5 & 6 & -2 \\ -12 & -5 & 6 \end{pmatrix};$ $B = \begin{pmatrix} 4 & 2 & -15 \\ 18 & 3 & 0,6 \end{pmatrix}$	$A = \begin{pmatrix} 5 & 6 & -2 \\ -12 & -5 & 6 \end{pmatrix};$ $Z = \begin{pmatrix} 4 & -12 \\ 36 & 5 \\ 46 & 0,6 \end{pmatrix}$	$N=6$	$(A-B)Z^2(A+B)^T$	$[4 \ 5 \ -6] \begin{pmatrix} 5 & 6 & 2 \\ -12 & 5 & -6 \\ 6 & 5 & -2 \end{pmatrix} \begin{bmatrix} -4 \\ 5 \\ 2 \end{bmatrix}$	$\begin{cases} 4,5x_1 + 5,6x_2 - 0,6x_3 = 5,6 \\ 0,2x_1 + 3,6x_2 + 12x_3 = 3,6 \\ -0,5x_1 + 1,4x_2 + 8,6x_3 = 4,8 \end{cases}$
2	$A = \begin{pmatrix} 4 & 5 & -6 \\ -14 & -6 & 5 \end{pmatrix};$ $B = \begin{pmatrix} 3 & 5 & -12 \\ 12 & 5 & 0,5 \end{pmatrix}$	$A = \begin{pmatrix} 4 & 5 & -6 \\ -14 & -6 & 5 \end{pmatrix};$ $Z = \begin{pmatrix} 2 & -8 \\ 25 & 6 \\ 46 & 0,4 \end{pmatrix}$	$N=4$	$(A+B)Z^4(A+B)^T$	$[4 \ 2 \ -15] \begin{pmatrix} 5 & 4 & 0,6 \\ -12 & 5 & -4 \\ 5 & 6 & -2 \end{pmatrix} \begin{bmatrix} -4 \\ 6 \\ 2 \end{bmatrix}$	$\begin{cases} 1,2x_1 + 4,6x_2 - 0,4x_3 = 3,6 \\ 0,5x_1 + 3,6x_2 + 14x_3 = 7,6 \\ -0,6x_1 + 3,4x_2 + 4,5x_3 = 1,8 \end{cases}$
3	$A = \begin{pmatrix} 2 & 4 & -5 \\ -14 & -5 & 6 \end{pmatrix};$ $B = \begin{pmatrix} 4 & 5 & -15 \\ 12 & 4 & 0,6 \end{pmatrix}$	$A = \begin{pmatrix} 2 & 4 & -5 \\ -14 & -5 & 6 \end{pmatrix};$ $Z = \begin{pmatrix} 6 & -15 \\ 35 & 5 \\ 48 & 0,4 \end{pmatrix}$	$N=5$	$(A+B)Z^4(A-B)^T$	$[4 \ 2 \ -15] \begin{pmatrix} 6 & 5 & 0 \\ -12 & 5 & -2 \\ 8 & 6 & -5 \end{pmatrix} \begin{bmatrix} -3 \\ 5 \\ 2 \end{bmatrix}$	$\begin{cases} 4,5x_1 + 0,6x_2 - 1,5x_3 = 4,6 \\ 3,6x_1 + 4,6x_2 + 5x_3 = 7,6 \\ -0,2x_1 + 1,4x_2 + 1,2x_3 = 3,8 \end{cases}$

4	$A = \begin{pmatrix} 4 & 5 & -2 \\ -12 & -5 & 6 \end{pmatrix};$ $B = \begin{pmatrix} 2 & 4 & -15 \\ 12 & 6 & 0.4 \end{pmatrix}$	$A = \begin{pmatrix} 4 & 5 & -2 \\ -12 & -5 & 6 \end{pmatrix};$ $Z = \begin{pmatrix} 15 & -14 \\ 15 & 4.2 \\ 46 & 0.5 \end{pmatrix}$	$N=4$	$(A-B)Z^4(A-B)^T$	$[2 \ 4 \ -15] \begin{pmatrix} 6 & 4 & 0 \\ -13 & 5 & -2 \\ 8 & 2 & -5 \end{pmatrix} \begin{bmatrix} -4 \\ 5 \\ 2 \end{bmatrix}$	$\begin{cases} 1.5x_1 + 4.6x_2 - 1.5x_3 = 3.6 \\ 4.6x_1 + 3.6x_2 + 12x_3 = 8.6 \\ -0.6x_1 + 3.6x_2 + 4.5x_3 = 5.4 \end{cases}$
5	$A = \begin{pmatrix} 7 & 3 & -5 \\ -12 & -9 & 2 \end{pmatrix};$ $B = \begin{pmatrix} 3 & 5 & -13 \\ 11 & 8 & 0.4 \end{pmatrix}$	$A = \begin{pmatrix} 7 & 3 & -5 \\ -12 & -9 & 2 \end{pmatrix};$ $Z = \begin{pmatrix} 3 & -11 \\ 33 & 5 \\ 47 & 0.6 \end{pmatrix}$	$N=2$	$(A-B)Z^3(A+B)^T$	$[4 \ 6 \ -10] \begin{pmatrix} 10 & 9 & 0.4 \\ -11 & 2 & -3 \\ 6 & 7 & -8 \end{pmatrix} \begin{bmatrix} -5 \\ 6 \\ 7 \end{bmatrix}$	$\begin{cases} 1.7x_1 + 3.5x_2 - 0.9x_3 = 4.7 \\ 0.7x_1 + 5.3x_2 + 13x_3 = 7.3 \\ -0.3x_1 + 5.1x_2 + 8.6x_3 = 5.4 \end{cases}$
6	$A = \begin{pmatrix} 1 & 4 & -6 \\ -12 & -5 & 7 \end{pmatrix};$ $B = \begin{pmatrix} 3 & 5 & -11 \\ 13 & 7 & 0.6 \end{pmatrix}$	$A = \begin{pmatrix} 1 & 4 & -6 \\ -12 & -5 & 7 \end{pmatrix};$ $Z = \begin{pmatrix} 3 & -12 \\ 37 & 6 \\ 43 & 0.5 \end{pmatrix}$	$N=3$	$(A-B)Z^2(A+B)^T$	$[4 \ 5 \ -12] \begin{pmatrix} 6 & 8 & 0.6 \\ -14 & 6 & -8 \\ 1 & 2 & -3 \end{pmatrix} \begin{bmatrix} -5 \\ 4 \\ 1 \end{bmatrix}$	$\begin{cases} 1.6x_1 + 3.1x_2 - 0.1x_3 = 4.6 \\ 0.9x_1 + 5.1x_2 + 11x_3 = 7.1 \\ -0.1x_1 + 5.1x_2 + 8.4x_3 = 5.1 \end{cases}$
7	$A = \begin{pmatrix} 6 & 7 & -9 \\ -14 & -3 & 5 \end{pmatrix};$ $B = \begin{pmatrix} 1 & 6 & -11 \\ 14 & 7 & 0.4 \end{pmatrix}$	$A = \begin{pmatrix} 6 & 7 & -9 \\ -14 & -3 & 5 \end{pmatrix};$ $Z = \begin{pmatrix} 3 & -12 \\ 31 & 3 \\ 41 & 0.5 \end{pmatrix}$	4	$(A+B)Z^2(A+B)^T$	$[3 \ 2 \ -11] \begin{pmatrix} 3 & 6 & 0.2 \\ -11 & 3 & -1 \\ 2 & 1 & -1 \end{pmatrix} \begin{bmatrix} -3 \\ 1 \\ 2 \end{bmatrix}$	$\begin{cases} 1.9x_1 + 3.2x_2 - 0.2x_3 = 4.2 \\ 0.8x_1 + 5.2x_2 + 12x_3 = 7.2 \\ -0.4x_1 + 5.6x_2 + 8.2x_3 = 5.2 \end{cases}$

8	$A = \begin{pmatrix} 4 & 3 & -5 \\ -11 & -2 & 5 \end{pmatrix};$ $B = \begin{pmatrix} 3 & 5 & -13 \\ 13 & 5 & 0.1 \end{pmatrix}$	$A = \begin{pmatrix} 4 & 3 & -5 \\ -11 & -2 & 5 \end{pmatrix};$ $Z = \begin{pmatrix} 4 & -14 \\ 32 & 7 \\ 42 & 0.2 \end{pmatrix}$	$N=6$	$(A+B)Z^2(A-B)^T$	$[2 \ 3 \ -13] \begin{pmatrix} 4 & 5 & 0.3 \\ -12 & 4 & -2 \\ 3 & 2 & -2 \end{pmatrix} \begin{bmatrix} -4 \\ 2 \\ 3 \end{bmatrix}$	$\begin{cases} 1.8x_1 + 3.3x_2 - 0.3x_3 = 4.3 \\ 0.7x_1 + 5.3x_2 + 13x_3 = 7.3 \\ -0.3x_1 + 5.3x_2 + 8.3x_3 = 5.3 \end{cases}$
9	$A = \begin{pmatrix} 5 & 4 & -6 \\ -15 & -3 & 6 \end{pmatrix};$ $B = \begin{pmatrix} 4 & 6 & -13 \\ 12 & 4 & 0.3 \end{pmatrix}$	$A = \begin{pmatrix} 5 & 4 & -6 \\ -15 & -3 & 6 \end{pmatrix};$ $Z = \begin{pmatrix} 5 & -12 \\ 33 & 2 \\ 43 & 0.4 \end{pmatrix}$	$N=5$	$(A+B)Z^2(A+B)^T$	$[4 \ 5 \ -14] \begin{pmatrix} 5 & 4 & 0.4 \\ -13 & 5 & -3 \\ 4 & 3 & -3 \end{pmatrix} \begin{bmatrix} -5 \\ 3 \\ 4 \end{bmatrix}$	$\begin{cases} 1.2x_1 + 3.4x_2 - 0.4x_3 = 4.4 \\ 0.6x_1 + 5.4x_2 + 14x_3 = 7.4 \\ -0.4x_1 + 5.4x_2 + 8.4x_3 = 5.4 \end{cases}$
10	$A = \begin{pmatrix} 6 & 5 & -7 \\ -11 & -4 & 7 \end{pmatrix};$ $B = \begin{pmatrix} 5 & 7 & -12 \\ 11 & 3 & 0.5 \end{pmatrix}$	$A = \begin{pmatrix} 6 & 5 & -7 \\ -11 & -4 & 7 \end{pmatrix};$ $Z = \begin{pmatrix} 6 & -13 \\ 34 & 3 \\ 44 & 0.6 \end{pmatrix}$	$N=3$	$(A-B)Z^4(A+B)^T$	$[3 \ 4 \ -15] \begin{pmatrix} 6 & 3 & 0.5 \\ -14 & 6 & -4 \\ 5 & 4 & -5 \end{pmatrix} \begin{bmatrix} -6 \\ 4 \\ 5 \end{bmatrix}$	$\begin{cases} 1.1x_1 + 3.5x_2 - 0.5x_3 = 4.5 \\ 0.5x_1 + 5.6x_2 + 15x_3 = 7.5 \\ -0.5x_1 + 5.6x_2 + 8.5x_3 = 5.6 \end{cases}$
11	$A = \begin{pmatrix} 4 & 6 & -8 \\ -14 & -5 & 8 \end{pmatrix};$ $B = \begin{pmatrix} 6 & 8 & -17 \\ 12 & 4 & 0.7 \end{pmatrix}$	$A = \begin{pmatrix} 4 & 6 & -8 \\ -14 & -5 & 8 \end{pmatrix};$ $Z = \begin{pmatrix} 7 & -15 \\ 35 & 4 \\ 41 & 0.8 \end{pmatrix}$	$N=2$	$(A+B)Z^4(A+B)^T$	$[2 \ 6 \ -17] \begin{pmatrix} 7 & 2 & 0.6 \\ -15 & 7 & -5 \\ 6 & 5 & -4 \end{pmatrix} \begin{bmatrix} -5 \\ 5 \\ 1 \end{bmatrix}$	$\begin{cases} 1.4x_1 + 3.6x_2 - 0.6x_3 = 4.6 \\ 0.4x_1 + 5.8x_2 + 14x_3 = 7.6 \\ -0.6x_1 + 5.8x_2 + 8.6x_3 = 5.8 \end{cases}$

12	$A = \begin{pmatrix} 5 & 7 & -9 \\ -14 & -6 & 9 \end{pmatrix};$ $B = \begin{pmatrix} 7 & 9 & -12 \\ 13 & 5 & 0.9 \end{pmatrix}$	$A = \begin{pmatrix} 5 & 7 & -9 \\ -14 & -6 & 9 \end{pmatrix};$ $Z = \begin{pmatrix} 8 & -11 \\ 36 & 5 \\ 42 & 0.1 \end{pmatrix}$	$N=4$	$(A-B)Z^4(A-B)^T$	$[5 \ 2 \ -18] \begin{pmatrix} 8 & 3 & 0.7 \\ -12 & 6 & -4 \\ 7 & 6 & -2 \end{pmatrix} \begin{bmatrix} -4 \\ 6 \\ 2 \end{bmatrix}$	$\begin{cases} 1.5x_1 + 3.7x_2 - 0.7x_3 = 4.7 \\ 0.3x_1 + 5.9x_2 + 13x_3 = 7.7 \\ -0.7x_1 + 5.9x_2 + 8.7x_3 = 5.9 \end{cases}$
13	$A = \begin{pmatrix} 6 & 8 & -6 \\ -10 & -7 & 6 \end{pmatrix};$ $B = \begin{pmatrix} 8 & 8 & -11 \\ 14 & 6 & 0.2 \end{pmatrix}$	$A = \begin{pmatrix} 6 & 8 & -6 \\ -10 & -7 & 6 \end{pmatrix};$ $Z = \begin{pmatrix} 4 & -12 \\ 37 & 4 \\ 43 & 0.3 \end{pmatrix}$	$N=5$	$(A+B)Z^3(A+B)^T$	$[3 \ 4 \ -11] \begin{pmatrix} 3 & 4 & 0.8 \\ -11 & 5 & -6 \\ 8 & 7 & -3 \end{pmatrix} \begin{bmatrix} -3 \\ 5 \\ 3 \end{bmatrix}$	$\begin{cases} 1.3x_1 + 3.8x_2 - 0.8x_3 = 4.8 \\ 0.2x_1 + 5.8x_2 + 12x_3 = 7.8 \\ -0.8x_1 + 5.8x_2 + 8.8x_3 = 5.6 \end{cases}$
14	$A = \begin{pmatrix} 4 & 9 & -7 \\ -13 & -8 & 5 \end{pmatrix};$ $B = \begin{pmatrix} 7 & 6 & -14 \\ 13 & 4 & 0.4 \end{pmatrix}$	$A = \begin{pmatrix} 4 & 9 & -7 \\ -13 & -8 & 5 \end{pmatrix};$ $Z = \begin{pmatrix} 5 & -15 \\ 38 & 5 \\ 44 & 0.5 \end{pmatrix}$	$N=4$	$(A-B)Z^3(A+B)^T$	$[4 \ 3 \ -13] \begin{pmatrix} 4 & 5 & 0.9 \\ -10 & 4 & -5 \\ 7 & 6 & -4 \end{pmatrix} \begin{bmatrix} -2 \\ 4 \\ 5 \end{bmatrix}$	$\begin{cases} 1.2x_1 + 3.9x_2 - 0.9x_3 = 4.9 \\ 0.3x_1 + 5.6x_2 + 11x_3 = 7.9 \\ -0.9x_1 + 5.6x_2 + 8.2x_3 = 5.4 \end{cases}$
15	$A = \begin{pmatrix} 5 & 6 & -8 \\ -14 & -9 & 4 \end{pmatrix};$ $B = \begin{pmatrix} 6 & 5 & -13 \\ 12 & 5 & 0.6 \end{pmatrix}$	$A = \begin{pmatrix} 5 & 6 & -8 \\ -14 & -9 & 4 \end{pmatrix};$ $Z = \begin{pmatrix} 5 & -10 \\ 35 & 4 \\ 45 & 0 \end{pmatrix}$	$N=6$	$(A+B)Z^3(A-B)^T$	$[5 \ 6 \ -14] \begin{pmatrix} 5 & 6 & 0.1 \\ -15 & 2 & -2 \\ 6 & 4 & -5 \end{pmatrix} \begin{bmatrix} -6 \\ 2 \\ 6 \end{bmatrix}$	$\begin{cases} 1.1x_1 + 3.5x_2 - 0.4x_3 = 4.5 \\ 0.4x_1 + 5.4x_2 + 10x_3 = 7.5 \\ -0.8x_1 + 5.4x_2 + 8.6x_3 = 5.8 \end{cases}$

Варианты к практическому занятию №5

№ варианта	№ пункта задания						
	1, 4	2	3	5	6	7	8
1	$A = \begin{pmatrix} 3 & 4 \\ 5 & 6 \end{pmatrix}, B = \begin{pmatrix} 7 & 8 \\ 9 & 10 \end{pmatrix},$ $C = \begin{pmatrix} 11 & 12 \\ 13 & 14 \end{pmatrix}, D = \begin{pmatrix} 15 & 16 \\ 17 & 18 \end{pmatrix}$	$Z = \begin{pmatrix} 3 & 4 \\ 5 & 6 \end{pmatrix}, a = \begin{pmatrix} 7 \\ 9 \end{pmatrix},$ $b = (-11 \ 17), N=4.6$	$E = \begin{pmatrix} \boxed{\begin{smallmatrix} \cdot & \cdot \\ \cdot & \cdot \end{smallmatrix}} & \boxed{\begin{smallmatrix} \cdot & \cdot \\ \cdot & \cdot \end{smallmatrix}} \\ \boxed{\begin{smallmatrix} \cdot & \cdot \\ \cdot & \cdot \end{smallmatrix}} & \begin{smallmatrix} \cdot & \cdot \\ \cdot & \cdot \end{smallmatrix} \end{pmatrix}$	$n=4,$ $l=5$	$l=6$	$n=5$	$a = -5, b = 7,$ $c = 5, e = 6$
2	$A = \begin{pmatrix} 4 & 5 \\ 6 & 7 \end{pmatrix}, B = \begin{pmatrix} 8 & 9 \\ 10 & 11 \end{pmatrix},$ $C = \begin{pmatrix} 12 & 13 \\ 14 & 15 \end{pmatrix}, D = \begin{pmatrix} 13 & 17 \\ 18 & 19 \end{pmatrix}$	$Z = \begin{pmatrix} 4 & 5 \\ 6 & 7 \end{pmatrix}, a = \begin{pmatrix} 8 \\ 10 \end{pmatrix},$ $b = (-12 \ 17), N=5.6$	$E = \begin{pmatrix} \boxed{\begin{smallmatrix} \cdot & \cdot \\ \cdot & \cdot \end{smallmatrix}} & \begin{smallmatrix} \cdot & \cdot \\ \cdot & \cdot \end{smallmatrix} \\ \begin{smallmatrix} \cdot & \cdot \\ \cdot & \cdot \end{smallmatrix} & \boxed{\begin{smallmatrix} \cdot & \cdot \\ \cdot & \cdot \end{smallmatrix}} \end{pmatrix}$	$n=5,$ $l=6$	$l=7$	$n=6$	$a = -6, b = 8,$ $c = 6, e = 7$
3	$A = \begin{pmatrix} 5 & 6 \\ 7 & 8 \end{pmatrix}, B = \begin{pmatrix} 9 & 10 \\ 11 & 12 \end{pmatrix},$ $C = \begin{pmatrix} 13 & 14 \\ 15 & 17 \end{pmatrix}, D = \begin{pmatrix} 17 & 18 \\ 19 & 20 \end{pmatrix}$	$Z = \begin{pmatrix} 5 & 6 \\ 7 & 8 \end{pmatrix}, a = \begin{pmatrix} 9 \\ 11 \end{pmatrix},$ $b = (-13 \ 18), N=7.6$	$E = \begin{pmatrix} \boxed{\begin{smallmatrix} \cdot & \cdot \\ \cdot & \cdot \end{smallmatrix}} & \begin{smallmatrix} \cdot & \cdot \\ \cdot & \cdot \end{smallmatrix} \\ \boxed{\begin{smallmatrix} \cdot & \cdot \\ \cdot & \cdot \end{smallmatrix}} & \begin{smallmatrix} \cdot & \cdot \\ \cdot & \cdot \end{smallmatrix} \end{pmatrix}$	$n=6,$ $l=7$	$l=8$	$n=7$	$a = -7, b = 9,$ $c = 7, e = 8$

4	$A = \begin{pmatrix} 6 & 9 \\ 8 & 9 \end{pmatrix}, B = \begin{pmatrix} 10 & 11 \\ 12 & 13 \end{pmatrix},$ $C = \begin{pmatrix} 14 & 15 \\ 17 & 18 \end{pmatrix}, D = \begin{pmatrix} 18 & 19 \\ 20 & 21 \end{pmatrix}$	$Z = \begin{pmatrix} 6 & 9 \\ 8 & 9 \end{pmatrix}, a = \begin{pmatrix} 10 \\ 11 \end{pmatrix},$ $b = (-14 \ 19), N=8.6$	$E = \begin{pmatrix} \begin{smallmatrix} \cdot & \cdot \\ \cdot & \cdot \end{smallmatrix} & \boxed{\begin{smallmatrix} \cdot & \cdot \\ \cdot & \cdot \end{smallmatrix}} \\ \begin{smallmatrix} \cdot & \cdot \\ \cdot & \cdot \end{smallmatrix} & \boxed{\begin{smallmatrix} \cdot & \cdot \\ \cdot & \cdot \end{smallmatrix}} \end{pmatrix}$	$n=4,$ $l=5$	$l=9$	$n=5$	$a = -8, b = 10,$ $c = 8, e = 9$
5	$A = \begin{pmatrix} 7 & 8 \\ 9 & 10 \end{pmatrix}, B = \begin{pmatrix} 11 & 12 \\ 13 & 14 \end{pmatrix},$ $C = \begin{pmatrix} 15 & 17 \\ 18 & 19 \end{pmatrix}, D = \begin{pmatrix} 19 & 20 \\ 21 & 22 \end{pmatrix}$	$Z = \begin{pmatrix} 7 & 8 \\ 9 & 10 \end{pmatrix}, a = \begin{pmatrix} 11 \\ 13 \end{pmatrix},$ $b = (-15 \ 20), N=9.6$	$E = \begin{pmatrix} \boxed{\begin{smallmatrix} \cdot & \cdot \\ \cdot & \cdot \end{smallmatrix}} & \boxed{\begin{smallmatrix} \cdot & \cdot \\ \cdot & \cdot \end{smallmatrix}} \\ \begin{smallmatrix} \cdot & \cdot \\ \cdot & \cdot \end{smallmatrix} & \begin{smallmatrix} \cdot & \cdot \\ \cdot & \cdot \end{smallmatrix} \end{pmatrix}$	$n=3,$ $l=6$	$l=7$	$n=6$	$a = -9, b = 10,$ $c = 9, e = 10$
6	$A = \begin{pmatrix} 8 & 9 \\ 10 & 11 \end{pmatrix}, B = \begin{pmatrix} 12 & 13 \\ 14 & 15 \end{pmatrix},$ $C = \begin{pmatrix} 15 & 17 \\ 18 & 19 \end{pmatrix}, D = \begin{pmatrix} 20 & 21 \\ 22 & 23 \end{pmatrix}$	$Z = \begin{pmatrix} 8 & 9 \\ 10 & 11 \end{pmatrix}, a = \begin{pmatrix} 12 \\ 14 \end{pmatrix},$ $b = (-17 \ 21), N=10.6$	$E = \begin{pmatrix} \begin{smallmatrix} \cdot & \cdot \\ \cdot & \cdot \end{smallmatrix} & \begin{smallmatrix} \cdot & \cdot \\ \cdot & \cdot \end{smallmatrix} \\ \boxed{\begin{smallmatrix} \cdot & \cdot \\ \cdot & \cdot \end{smallmatrix}} & \boxed{\begin{smallmatrix} \cdot & \cdot \\ \cdot & \cdot \end{smallmatrix}} \end{pmatrix}$	$n=5,$ $l=6$	$l=8$	$n=4$	$a = -10, b = 12,$ $c = 10, e = 11$
7	$A = \begin{pmatrix} 9 & 10 \\ 11 & 12 \end{pmatrix}, B = \begin{pmatrix} 13 & 14 \\ 15 & 17 \end{pmatrix},$ $C = \begin{pmatrix} 17 & 18 \\ 19 & 20 \end{pmatrix}, D = \begin{pmatrix} 21 & 22 \\ 23 & 24 \end{pmatrix}$	$Z = \begin{pmatrix} 9 & 10 \\ 11 & 12 \end{pmatrix}, a = \begin{pmatrix} 13 \\ 15 \end{pmatrix},$ $b = (-17 \ 22), N=8.6$	$E = \begin{pmatrix} \begin{smallmatrix} \cdot & \cdot \\ \cdot & \cdot \end{smallmatrix} & \boxed{\begin{smallmatrix} \cdot & \cdot \\ \cdot & \cdot \end{smallmatrix}} \\ \boxed{\begin{smallmatrix} \cdot & \cdot \\ \cdot & \cdot \end{smallmatrix}} & \begin{smallmatrix} \cdot & \cdot \\ \cdot & \cdot \end{smallmatrix} \end{pmatrix}$	$n=4,$ $l=7$	$l=9$	$n=6$	$a = -11, b = 13,$ $c = 11, e = 12$

8	$A = \begin{pmatrix} 10 & 11 \\ 12 & 13 \end{pmatrix}, B = \begin{pmatrix} 14 & 15 \\ 15 & 17 \end{pmatrix},$ $C = \begin{pmatrix} 18 & 19 \\ 20 & 21 \end{pmatrix}, D = \begin{pmatrix} 22 & 23 \\ 24 & 25 \end{pmatrix}$	$Z = \begin{pmatrix} 10 & 11 \\ 12 & 13 \end{pmatrix}, a = \begin{pmatrix} 14 \\ 17 \end{pmatrix},$ $b = (-18 \ 24), N=7.6$	$E = \begin{pmatrix} \boxed{\begin{smallmatrix} \cdot & \cdot \\ \cdot & \cdot \end{smallmatrix}} & \begin{smallmatrix} \cdot & \cdot \\ \cdot & \cdot \end{smallmatrix} \\ \begin{smallmatrix} \cdot & \cdot \\ \cdot & \cdot \end{smallmatrix} & \boxed{\begin{smallmatrix} \cdot & \cdot \\ \cdot & \cdot \end{smallmatrix}} \end{pmatrix}$	$n=3,$ $l=4$	$l=10$	$n=5$	$a = -12, b = 14,$ $c = 12, e = 13$
9	$A = \begin{pmatrix} 11 & 12 \\ 13 & 14 \end{pmatrix}, B = \begin{pmatrix} 15 & 17 \\ 18 & 20 \end{pmatrix},$ $C = \begin{pmatrix} 19 & 20 \\ 21 & 22 \end{pmatrix}, D = \begin{pmatrix} 23 & 24 \\ 25 & 26 \end{pmatrix}$	$Z = \begin{pmatrix} 11 & 12 \\ 13 & 14 \end{pmatrix}, a = \begin{pmatrix} 15 \\ 17 \end{pmatrix},$ $b = (-19 \ 24), N=5.6$	$E = \begin{pmatrix} \boxed{\begin{smallmatrix} \cdot & \cdot \\ \cdot & \cdot \end{smallmatrix}} & \begin{smallmatrix} \cdot & \cdot \\ \cdot & \cdot \end{smallmatrix} \\ \boxed{\begin{smallmatrix} \cdot & \cdot \\ \cdot & \cdot \end{smallmatrix}} & \begin{smallmatrix} \cdot & \cdot \\ \cdot & \cdot \end{smallmatrix} \end{pmatrix}$	$n=4,$ $l=5$	$l=11$	$n=4$	$a = -13, b = 15,$ $c = 13, e = 14$
10	$A = \begin{pmatrix} 12 & 12 \\ 14 & 15 \end{pmatrix}, B = \begin{pmatrix} 17 & 18 \\ 19 & 20 \end{pmatrix},$ $C = \begin{pmatrix} 20 & 21 \\ 22 & 23 \end{pmatrix}, D = \begin{pmatrix} 24 & 25 \\ 26 & 27 \end{pmatrix}$	$Z = \begin{pmatrix} 12 & 12 \\ 14 & 15 \end{pmatrix}, a = \begin{pmatrix} 17 \\ 18 \end{pmatrix},$ $b = (-20 \ 25), N=4.6$	$E = \begin{pmatrix} \begin{smallmatrix} \cdot & \cdot \\ \cdot & \cdot \end{smallmatrix} & \boxed{\begin{smallmatrix} \cdot & \cdot \\ \cdot & \cdot \end{smallmatrix}} \\ \begin{smallmatrix} \cdot & \cdot \\ \cdot & \cdot \end{smallmatrix} & \boxed{\begin{smallmatrix} \cdot & \cdot \\ \cdot & \cdot \end{smallmatrix}} \end{pmatrix}$	$n=5,$ $l=6$	$l=12$	$n=6$	$a = -14, b = 17,$ $c = 15, e = 18$
11	$A = \begin{pmatrix} 13 & 14 \\ 15 & 17 \end{pmatrix}, B = \begin{pmatrix} 17 & 18 \\ 19 & 20 \end{pmatrix},$ $C = \begin{pmatrix} 21 & 22 \\ 23 & 24 \end{pmatrix}, D = \begin{pmatrix} 25 & 26 \\ 27 & 28 \end{pmatrix}$	$Z = \begin{pmatrix} 13 & 14 \\ 15 & 17 \end{pmatrix}, a = \begin{pmatrix} 17 \\ 19 \end{pmatrix},$ $b = (-21 \ 26), N=3.6$	$E = \begin{pmatrix} \boxed{\begin{smallmatrix} \cdot & \cdot \\ \cdot & \cdot \end{smallmatrix}} & \boxed{\begin{smallmatrix} \cdot & \cdot \\ \cdot & \cdot \end{smallmatrix}} \\ \begin{smallmatrix} \cdot & \cdot \\ \cdot & \cdot \end{smallmatrix} & \begin{smallmatrix} \cdot & \cdot \\ \cdot & \cdot \end{smallmatrix} \end{pmatrix}$	$n=3,$ $l=6$	$l=10$	$n=5$	$a = -15, b = 17,$ $c = 15, e = 18$

12	$A = \begin{pmatrix} 14 & 15 \\ 17 & 19 \end{pmatrix}, B = \begin{pmatrix} 18 & 19 \\ 20 & 21 \end{pmatrix},$ $C = \begin{pmatrix} 22 & 23 \\ 24 & 25 \end{pmatrix}, D = \begin{pmatrix} 17 & 19 \\ 20 & 21 \end{pmatrix}$	$Z = \begin{pmatrix} 14 & 15 \\ 17 & 19 \end{pmatrix}, a = \begin{pmatrix} 18 \\ 20 \end{pmatrix},$ $b = (-22 \ 27), N=4.6$	$E = \begin{pmatrix} \begin{smallmatrix} \cdot & \cdot \\ \cdot & \cdot \end{smallmatrix} & \begin{smallmatrix} \cdot & \cdot \\ \cdot & \cdot \end{smallmatrix} \\ \begin{smallmatrix} \cdot & \cdot \\ \cdot & \cdot \end{smallmatrix} & \begin{smallmatrix} \cdot & \cdot \\ \cdot & \cdot \end{smallmatrix} \end{pmatrix}$	$n=4,$ $l=5$	$l=9$	$n=6$	$a = -17, b = 19,$ $c = 19, e = 20$
13	$A = \begin{pmatrix} 15 & 17 \\ 18 & 19 \end{pmatrix}, B = \begin{pmatrix} 19 & 20 \\ 21 & 22 \end{pmatrix},$ $C = \begin{pmatrix} 23 & 24 \\ 25 & 26 \end{pmatrix}, D = \begin{pmatrix} 17 & 18 \\ 19 & 20 \end{pmatrix}$	$Z = \begin{pmatrix} 15 & 17 \\ 18 & 19 \end{pmatrix}, a = \begin{pmatrix} 19 \\ 21 \end{pmatrix},$ $b = (-23 \ 29), N=5.6$	$E = \begin{pmatrix} \begin{smallmatrix} \cdot & \cdot \\ \cdot & \cdot \end{smallmatrix} & \begin{smallmatrix} \cdot & \cdot \\ \cdot & \cdot \end{smallmatrix} \\ \begin{smallmatrix} \cdot & \cdot \\ \cdot & \cdot \end{smallmatrix} & \begin{smallmatrix} \cdot & \cdot \\ \cdot & \cdot \end{smallmatrix} \end{pmatrix}$	$n=5,$ $l=6$	$l=11$	$n=5$	$a = -17, b = 19,$ $c = 17, e = 18$
14	$A = \begin{pmatrix} 17 & 18 \\ 19 & 20 \end{pmatrix}, B = \begin{pmatrix} 20 & 21 \\ 22 & 23 \end{pmatrix},$ $C = \begin{pmatrix} 24 & 25 \\ 26 & 27 \end{pmatrix}, D = \begin{pmatrix} 29 & 30 \\ 31 & 32 \end{pmatrix}$	$Z = \begin{pmatrix} 17 & 18 \\ 19 & 20 \end{pmatrix}, a = \begin{pmatrix} 20 \\ 22 \end{pmatrix},$ $b = (-24 \ 29), N=9.6$	$E = \begin{pmatrix} \begin{smallmatrix} \cdot & \cdot \\ \cdot & \cdot \end{smallmatrix} & \begin{smallmatrix} \cdot & \cdot \\ \cdot & \cdot \end{smallmatrix} \\ \begin{smallmatrix} \cdot & \cdot \\ \cdot & \cdot \end{smallmatrix} & \begin{smallmatrix} \cdot & \cdot \\ \cdot & \cdot \end{smallmatrix} \end{pmatrix}$	$n=3,$ $l=5$	$l=10$	$n=7$	$a = -18, b = 20,$ $c = 18, e = 19$
15	$A = \begin{pmatrix} 17 & 18 \\ 19 & 20 \end{pmatrix}, B = \begin{pmatrix} 21 & 22 \\ 23 & 24 \end{pmatrix},$ $C = \begin{pmatrix} 25 & 26 \\ 27 & 32 \end{pmatrix}, D = \begin{pmatrix} 29 & 30 \\ 31 & 34 \end{pmatrix}$	$Z = \begin{pmatrix} 17 & 18 \\ 19 & 20 \end{pmatrix}, a = \begin{pmatrix} 21 \\ 23 \end{pmatrix},$ $b = (-10 \ 15), N=3.6$	$E = \begin{pmatrix} \begin{smallmatrix} \cdot & \cdot \\ \cdot & \cdot \end{smallmatrix} & \begin{smallmatrix} \cdot & \cdot \\ \cdot & \cdot \end{smallmatrix} \\ \begin{smallmatrix} \cdot & \cdot \\ \cdot & \cdot \end{smallmatrix} & \begin{smallmatrix} \cdot & \cdot \\ \cdot & \cdot \end{smallmatrix} \end{pmatrix}$	$n=4,$ $l=6$	$l=8$	$n=6$	$a = -19, b = 21,$ $c = 19, e = 20$

Варианты к практическому занятию №6

№ варианта	№ пункта задания	
	1	2
1	$A = \begin{pmatrix} 1 & -2 & -3 \\ 2 & -1 & 15 \\ 2 & -1 & 2 \end{pmatrix}$	$N=4$
2	$A = \begin{pmatrix} 2 & -3 & -4 \\ 3 & -2 & 14 \\ 3 & -2 & 3 \end{pmatrix}$	$N=5$
3	$A = \begin{pmatrix} 4 & -5 & -6 \\ 4 & -3 & 13 \\ 4 & -3 & 4 \end{pmatrix}$	$N=6$
4	$A = \begin{pmatrix} 5 & -6 & -7 \\ 5 & -4 & 12 \\ 5 & -4 & 5 \end{pmatrix}$	$N=7$
5	$A = \begin{pmatrix} 6 & -7 & -8 \\ 6 & -5 & 11 \\ 9 & -5 & 2 \end{pmatrix}$	$N=8$
6	$A = \begin{pmatrix} 7 & -8 & -9 \\ 7 & -6 & 10 \\ 5 & -6 & 3 \end{pmatrix}$	$N=7$
7	$A = \begin{pmatrix} 8 & -9 & -7 \\ 8 & -7 & 15 \\ 9 & -7 & 4 \end{pmatrix}$	$N=6$
8	$A = \begin{pmatrix} 9 & -7 & -6 \\ 9 & -8 & 14 \\ 4 & -8 & 5 \end{pmatrix}$	$N=5$
9	$A = \begin{pmatrix} 4 & -6 & -5 \\ 7 & -9 & 13 \\ 3 & -9 & 6 \end{pmatrix}$	$N=4$
10	$A = \begin{pmatrix} 5 & -2 & -4 \\ 6 & -7 & 12 \\ 4 & -7 & 8 \end{pmatrix}$	$N=6$
11	$A = \begin{pmatrix} 6 & -4 & -3 \\ 5 & -6 & 11 \\ 5 & -4 & 8 \end{pmatrix}$	$N=7$

12	$A = \begin{pmatrix} 7 & -3 & -2 \\ 4 & -5 & 10 \\ 6 & -8 & 9 \end{pmatrix}$	$N=8$
13	$A = \begin{pmatrix} 8 & -2 & -1 \\ 3 & -4 & 11 \\ 7 & -4 & 8 \end{pmatrix}$	$N=4$
14	$A = \begin{pmatrix} 9 & -5 & -7 \\ 2 & -3 & 12 \\ 8 & -5 & 6 \end{pmatrix}$	$N=5$
15	$A = \begin{pmatrix} 4 & -6 & -8 \\ 1 & -2 & 15 \\ 9 & -5 & 6 \end{pmatrix}$	$N=6$

Варианты к практическому занятию №7

№ варианта	№ пункта задания		
	1	2	3 - 4
1	$N=11, a=4, b=8$	$n=100$	$n=2, l=11, a=0.1\pi, b=5\pi, a_5=0.1, b_5=5$
2	$N=12, a=5, b=9$	$n=110$	$n=3, l=12, a=0.2\pi, b=6\pi, a_5=0.2, b_5=6$
3	$N=13, a=6, b=20$	$n=120$	$n=4, l=13, a=0.3\pi, b=7\pi, a_5=0.3, b_5=7$
4	$N=14, a=7, b=21$	$n=130$	$n=3, l=14, a=0.4\pi, b=8\pi, a_5=0.4, b_5=8$
5	$N=15, a=8, b=22$	$n=140$	$n=4, l=15, a=0.5\pi, b=9\pi, a_5=0.5, b_5=9$
6	$N=17, a=9, b=23$	$n=150$	$n=2, l=11, a=0.1\pi, b=5\pi, a_5=0.1, b_5=5$
7	$N=18, a=10, b=46$	$n=170$	$n=3, l=12, a=0.2\pi, b=6\pi, a_5=0.2, b_5=6$
8	$N=19, a=4, b=25$	$n=180$	$n=4, l=13, a=0.3\pi, b=7\pi, a_5=0.3, b_5=7$
9	$N=20, a=5, b=26$	$n=190$	$n=2, l=14, a=0.4\pi, b=8\pi, a_5=0.4, b_5=8$
10	$N=21, a=6, b=32$	$n=110$	$n=3, l=15, a=0.5\pi, b=9\pi, a_5=0.5, b_5=9$
11	$N=22, a=7, b=33$	$n=120$	$n=4, l=11, a=0.1\pi, b=5\pi, a_5=0.1, b_5=5$
12	$N=23, a=8, b=34$	$n=130$	$n=2, l=12, a=0.2\pi, b=6\pi, a_5=0.2, b_5=6$
13	$N=24, a=9, b=35$	$n=140$	$n=3, l=13, a=0.3\pi, b=7\pi, a_5=0.3, b_5=7$
14	$N=25, a=10, b=36$	$n=150$	$n=4, l=14, a=0.4\pi, b=8\pi, a_5=0.4, b_5=8$
15	$N=26, a=5, b=48$	$n=200$	$n=2, l=15, a=0.5\pi, b=9\pi, a_5=0.5, b_5=9$

Варианты к практическому занятию №8

№ варианта	№ пункта задания	
	1	2
1	$y(x) = \begin{cases} \cos(x), & x \leq -\pi; \\ x/\pi, & -\pi < x < \pi; \quad x \in [-2\pi, 2\pi] \\ \sin(x), & x \geq \pi \end{cases}$	$y(x) = \begin{cases} \cos(x), & x = -\pi; \\ x/\pi, & x = \pi; \quad x = -\pi, \pi, 2\pi \\ \sin(x), & x \neq -\pi \wedge x \neq \pi \end{cases}$
2	$y(x) = \begin{cases} \sin(x), & x \leq -\pi; \\ 2x/\pi, & -\pi < x < \pi; \quad x \in [-2\pi, 2\pi] \\ \cos(x), & x \geq \pi \end{cases}$	$y(x) = \begin{cases} \sin(x), & x = -\pi; \\ 2x/\pi, & x = \pi; \quad x = -\pi, \pi, 2\pi \\ \cos(x), & x \neq -\pi \wedge x \neq \pi \end{cases}$
3	$y(x) = \begin{cases} \cos(x), & x \leq -\pi; \\ x/2\pi, & -\pi < x < \pi; \quad x \in [-2\pi, 2\pi] \\ \sin(x), & x \geq \pi \end{cases}$	$y(x) = \begin{cases} \cos(x), & x = -\pi; \\ x/2\pi, & x = \pi; \quad x = -\pi, \pi, 2\pi \\ \sin(x), & x \neq -\pi \wedge x \neq \pi \end{cases}$
4	$y(x) = \begin{cases} tg(x), & x \leq -\pi; \\ x/\pi, & -\pi < x < \pi; \quad x \in [-2\pi, 2\pi] \\ ctg(x), & x \geq \pi \end{cases}$	$y(x) = \begin{cases} tg(x), & x = -\pi; \\ x/\pi, & x = \pi; \quad x = -\pi, \pi, 2\pi \\ ctg(x), & x \neq -\pi \wedge x \neq \pi \end{cases}$
5	$y(x) = \begin{cases} ctg(x), & x \leq -\pi; \\ x/\pi, & -\pi < x < \pi; \quad x \in [-2\pi, 2\pi] \\ tg(x), & x \geq \pi \end{cases}$	$y(x) = \begin{cases} ctg(x), & x = -\pi; \\ x/\pi, & x = \pi; \quad x = -\pi, \pi, 2\pi \\ tg(x), & x \neq -\pi \wedge x \neq \pi \end{cases}$
6	$y(x) = \begin{cases} tg(x), & x \leq -\pi; \\ 2x/\pi, & -\pi < x < \pi; \quad x \in [-2\pi, 2\pi] \\ ctg(x), & x \geq \pi \end{cases}$	$y(x) = \begin{cases} tg(x), & x = -\pi; \\ 2x/\pi, & x = \pi; \quad x = -\pi, \pi, 2\pi \\ ctg(x), & x \neq -\pi \wedge x \neq \pi \end{cases}$
7	$y(x) = \begin{cases} ctg(x), & x \leq -\pi; \\ 2x/\pi, & -\pi < x < \pi; \quad x \in [-2\pi, 2\pi] \\ tg(x), & x \geq \pi \end{cases}$	$y(x) = \begin{cases} ctg(x), & x = -\pi; \\ 2x/\pi, & x = \pi; \quad x = -\pi, \pi, 2\pi \\ tg(x), & x \neq -\pi \wedge x \neq \pi \end{cases}$
8	$y(x) = \begin{cases} tg(x), & x \leq -\pi; \\ x/2\pi, & -\pi < x < \pi; \quad x \in [-2\pi, 2\pi] \\ ctg(x), & x \geq \pi \end{cases}$	$y(x) = \begin{cases} tg(x), & x = -\pi; \\ x/2\pi, & x = \pi; \quad x = -\pi, \pi, 2\pi \\ ctg(x), & x \neq -\pi \wedge x \neq \pi \end{cases}$
9	$y(x) = \begin{cases} ctg(x), & x \leq -\pi; \\ x/2\pi, & -\pi < x < \pi; \quad x \in [-2\pi, 2\pi] \\ tg(x), & x \geq \pi \end{cases}$	$y(x) = \begin{cases} ctg(x), & x = -\pi; \\ x/2\pi, & x = \pi; \quad x = -\pi, \pi, 2\pi \\ tg(x), & x \neq -\pi \wedge x \neq \pi \end{cases}$
10	$y(x) = \begin{cases} \exp(x), & x \leq -\pi; \\ x/\pi, & -\pi < x < \pi; \quad x \in [-2\pi, 2\pi] \\ \cos(x), & x \geq \pi \end{cases}$	$y(x) = \begin{cases} \exp(x), & x = -\pi; \\ x/\pi, & x = \pi; \quad x = -\pi, \pi, 2\pi \\ \cos(x), & x \neq -\pi \wedge x \neq \pi \end{cases}$
11	$y(x) = \begin{cases} \exp(x), & x \leq -\pi; \\ x/\pi, & -\pi < x < \pi; \quad x \in [-2\pi, 2\pi] \\ \sin(x), & x \geq \pi \end{cases}$	$y(x) = \begin{cases} \exp(x), & x = -\pi; \\ x/\pi, & x = \pi; \quad x = -\pi, \pi, 2\pi \\ \sin(x), & x \neq -\pi \wedge x \neq \pi \end{cases}$

12	$y(x) = \begin{cases} \exp(x), & x \leq -\pi; \\ x/\pi, & -\pi < x < \pi; \quad x \in [-2\pi, 2\pi] \\ \operatorname{tg}(x), & x \geq \pi \end{cases}$	$y(x) = \begin{cases} \exp(x), & x = -\pi; \\ x/\pi, & x = \pi; \quad x = -\pi, \pi, 2\pi \\ \operatorname{tg}(x), & x \neq -\pi \wedge x \neq \pi \end{cases}$
13	$y(x) = \begin{cases} \exp(x), & x \leq -\pi; \\ x/\pi, & -\pi < x < \pi; \quad x \in [-2\pi, 2\pi] \\ \operatorname{ctg}(x), & x \geq \pi \end{cases}$	$y(x) = \begin{cases} \exp(x), & x = -\pi; \\ x/\pi, & x = \pi; \quad x = -\pi, \pi, 2\pi \\ \operatorname{ctg}(x), & x \neq -\pi \wedge x \neq \pi \end{cases}$
14	$y(x) = \begin{cases} \sin(x), & x \leq -\pi; \\ x/\pi, & -\pi < x < \pi; \quad x \in [-2\pi, 2\pi] \\ \exp(x), & x \geq \pi \end{cases}$	$y(x) = \begin{cases} \sin(x), & x = -\pi; \\ x/\pi, & x = \pi; \quad x = -\pi, \pi, 2\pi \\ \exp(x), & x \neq -\pi \wedge x \neq \pi \end{cases}$
15	$y(x) = \begin{cases} \cos(x), & x \leq -\pi; \\ x/\pi, & -\pi < x < \pi; \quad x \in [-2\pi, 2\pi] \\ \exp(x), & x \geq \pi \end{cases}$	$y(x) = \begin{cases} \cos(x), & x = -\pi; \\ x/\pi, & x = \pi; \quad x = -\pi, \pi, 2\pi \\ \exp(x), & x \neq -\pi \wedge x \neq \pi \end{cases}$

Варианты к практическому занятию №9

№ варианта	№ пункта задания				
	1-2	3	4-5	6	7-8
1	$f(x) = \sin(x)e^x;$ $x \in [1, 2]$	$f(x) = \sin^2(x)e^{-0.1x};$ $\varphi(x) = \sin^2(x)e^{-0.2x};$ $x \in [-2\pi, 2\pi]$	$data=[35 \ 18 \ 5]$	$f(x) = \frac{1}{x^4};$ $\varphi(x) = 100(x+0.5)^{-5};$ $x \in [0.4, 5]$	$z(x, y) = 2 \sin(2\pi x) \cos(2.5\pi y)(1-x^2)y(1-y);$ $x \in [-2, 2]; y \in [0, 1]$
2	$f(x) = \operatorname{tg}(x)e^x;$ $x \in [1, 2]$	$f(x) = \cos^2(x)e^{-0.2x};$ $\varphi(x) = \cos^2(x)e^{-0.1x};$ $x \in [-2\pi, 2\pi]$	$data=[36 \ 19 \ 6]$	$f(x) = \frac{1}{x^6};$ $\varphi(x) = 100(x+0.4)^{-3};$ $x \in [0.2, 5]$	$z(x, y) = 3 \sin(3\pi x) \cos(3.5\pi y)(1-x^2)y(1-y);$ $x \in [-1, 1]; y \in [1, 2]$
3	$f(x) = \operatorname{ctg}(x)e^x;$ $x \in [1, 4]$	$f(x) = \sin^2(x)e^{-0.2x};$ $\varphi(x) = \sin^2(x)e^{-0.1x};$ $x \in [-2\pi, 2\pi]$	$data=[37 \ 20 \ 7]$	$f(x) = \frac{1}{x^2};$ $\varphi(x) = 100(x+0.6)^{-4};$ $x \in [0.2, 6]$	$z(x, y) = 4 \sin(4\pi x) \cos(4.5\pi y)(1-x^2)y(1-y);$ $x \in [-3, 3]; y \in [0, 1]$
4	$f(x) = \sin(x) \cos(x);$ $x \in [-1, 1]$	$f(x) = \tan(x)e^{-0.1x};$ $\varphi(x) = \tan(x)e^{-0.2x};$ $x \in [-\pi, \pi]$	$data=[38 \ 15 \ 8]$	$f(x) = \frac{1}{x^4};$ $\varphi(x) = 100(x+0.5)^{-6};$ $x \in [0.5, 8]$	$z(x, y) = 6 \sin(5\pi x) \cos(2.5\pi y)(1-x^2)y(1-y);$ $x \in [-1, 1]; y \in [2, 5]$

5	$f(x) = \sin(x)tg(x);$ $x \in [-2, -1]$	$f(x) = ctg^2(x)e^{-0.1x};$ $\varphi(x) = ctg^2(x)e^{-0.2x};$ $x \in [-\pi, \pi]$	$data=[39 \ 17 \ 9]$	$f(x) = \frac{1}{x^3};$ $\varphi(x) = 100(x+0.2)^{-8};$ $x \in [0.2, 5]$	$z(x, y) = 7 \sin(6\pi x) \cos(3.5\pi y)(1-x^2)y(1-y);$ $x \in [-2, 2]; y \in [0, 1]$
6	$f(x) = \sin(x)ctg(x);$ $x \in [-2, -1]$	$f(x) = tg^2(x)e^{-0.2x};$ $\varphi(x) = tg^2(x)e^{-0.1x};$ $x \in [-\pi, \pi]$	$data=[40 \ 18 \ 10]$	$f(x) = \frac{1}{x^2};$ $\varphi(x) = 100(x+0.4)^{-6};$ $x \in [1, 5]$	$z(x, y) = 8 \sin(2\pi x) \cos(4.5\pi y)(1-x^2)y(1-y);$ $x \in [-1, 1]; y \in [0, 2]$
7	$f(x) = \cos(x)tg(x);$ $x \in [-2, -1]$	$f(x) = \sin^2(x) \ln(-0.1x+1);$ $\varphi(x) = \sin^2(x) \ln(-0.2x+1);$ $x \in [-2\pi, 2\pi]$	$data=[25 \ 19 \ 11]$	$f(x) = \frac{1}{x^3};$ $\varphi(x) = 1000(x+0.5)^{-4};$ $x \in [0.5, 3]$	$z(x, y) = 9 \sin(3\pi x) \cos(2.5\pi y)(1-x^2)y(1-y);$ $x \in [-4, 4]; y \in [0, 1]$
8	$f(x) = \cos(x)ctg(x);$ $x \in [2, 4]$	$f(x) = \cos^2(x) \ln(-0.1x+1);$ $\varphi(x) = \cos^2(x) \ln(-0.2x+1);$ $x \in [-2\pi, 2\pi]$	$data=[26 \ 20 \ 12]$	$f(x) = \frac{1}{x^4};$ $\varphi(x) = 1000(x+0.5)^{-5};$ $x \in [0.2, 6]$	$z(x, y) = 10 \sin(4\pi x) \cos(1.5\pi y)(1-x^2)y(1-y);$ $x \in [-1, 1]; y \in [1, 5]$
9	$f(x) = \sin(x) \ln(x);$ $x \in [1, 4]$	$f(x) = tg^2(x) \ln(-0.1x+1);$ $\varphi(x) = tg^2(x) \ln(-0.2x+1);$ $x \in [-\pi, \pi]$	$data=[36 \ 14 \ 13]$	$f(x) = \frac{1}{x^5};$ $\varphi(x) = 1000(x+0.4)^{-6};$ $x \in [0.2, 8]$	$z(x, y) = 11 \sin(5\pi x) \cos(3.5\pi y)(1-x^2)y(1-y);$ $x \in [-5, 5]; y \in [0, 1]$
10	$f(x) = \cos(x) \ln(x);$ $x \in [1, 4]$	$f(x) = ctg^2(x) \ln(-0.1x+1);$ $\varphi(x) = ctg^2(x) \ln(-0.2x+1);$ $x \in [-2\pi, -\pi]$	$data=[38 \ 13 \ 14]$	$f(x) = \frac{1}{x^4};$ $\varphi(x) = 100(x+0.2)^{-6};$ $x \in [0.4, 6]$	$z(x, y) = 12 \sin(6\pi x) \cos(4.5\pi y)(1-x^2)y(1-y);$ $x \in [-1, 1]; y \in [0, 5]$

11	$f(x) = tg(x) \ln(x);$ $x \in [1, 2]$	$f(x) = \sin^2(x) \lg(-0.1x+1);$ $\varphi(x) = \sin^2(x) \lg(-0.2x+1);$ $x \in [-2\pi, 2\pi]$	$data=[39 \ 12 \ 15]$	$f(x) = \frac{1}{x^4};$ $\varphi(x) = 100(x+0.2)^{-5};$ $x \in [0.2, 5]$	$z(x, y) = 13 \sin(2\pi x) \cos(2.5\pi y)(1-x^2)y(1-y);$ $x \in [-4, 4]; y \in [0, 1]$
12	$f(x) = ctg(x) \ln(x);$ $x \in [1, 2]$	$f(x) = \cos^2(x) \lg(-0.1x+1);$ $\varphi(x) = \cos^2(x) \lg(-0.2x+1);$ $x \in [-2\pi, 2\pi]$	$data=[40 \ 12 \ 14]$	$f(x) = \frac{1}{x^6};$ $\varphi(x) = 100(x+0.5)^{-3};$ $x \in [0.5, 6]$	$z(x, y) = 14 \sin(3\pi x) \cos(1.5\pi y)(1-x^2)y(1-y);$ $x \in [-1, 1]; y \in [0, 4]$
13	$f(x) = \sin(x) \lg(x);$ $x \in [1, 4]$	$f(x) = tg^2(x) \lg(-0.1x+1);$ $\varphi(x) = tg^2(x) \lg(-0.2x+1);$ $x \in [-\pi, \pi]$	$data=[42 \ 14 \ 13]$	$f(x) = \frac{1}{x^2};$ $\varphi(x) = 100(x+0.2)^{-4};$ $x \in [0.5, 8]$	$z(x, y) = 15 \sin(4\pi x) \cos(3.5\pi y)(1-x^2)y(1-y);$ $x \in [-2, 2]; y \in [0, 1]$
14	$f(x) = \cos(x) \lg(x);$ $x \in [1, 4]$	$f(x) = ctg^2(x) \lg(-0.1x+1);$ $\varphi(x) = ctg^2(x) \lg(-0.2x+1);$ $x \in [-2\pi, -\pi]$	$data=[45 \ 18 \ 12]$	$f(x) = \frac{1}{x^4};$ $\varphi(x) = 100(x+0.2)^{-6};$ $x \in [0.2, 5]$	$z(x, y) = 4 \sin(5\pi x) \cos(4.5\pi y)(1-x^2)y(1-y);$ $x \in [-1, 1]; y \in [5, 6]$
15	$f(x) = \ln(x)e^x;$ $x \in [1, 4]$	$f(x) = \sin^2(x) \lg(-0.2x+5);$ $\varphi(x) = \sin^2(x) \lg(-0.1x+5);$ $x \in [-2\pi, 2\pi]$	$data=[46 \ 20 \ 15]$	$f(x) = \frac{1}{x^3};$ $\varphi(x) = 100(x+0.5)^{-8};$ $x \in [0.5, 6]$	$z(x, y) = 6 \sin(8\pi x) \cos(2.5\pi y)(1-x^2)y(1-y);$ $x \in [-5, 5]; y \in [0, 1]$





Варианты к практическому занятию №10

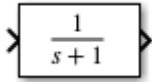
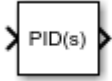



№ варианта	№ пункта задания	
	1-9	10
1	$A(p) = 2p + 2.5;$ $B(p) = 2p^2 + 3.5p + 5$	$W = \frac{2p+1}{p^2+p+1}, W_1 = \frac{3p+1}{4p^2+p+1}, W_2 = \frac{7p+1}{5p^2+p+1}$
2	$A(p) = 3p + 3.5;$ $B(p) = 3p^2 + 4.5p + 6$	$W = \frac{3p+1}{p^2+p+1}, W_1 = \frac{4p+1}{5p^2+p+1}, W_2 = \frac{8p+1}{6p^2+p+1}$
3	$A(p) = 4p + 4.5;$ $B(p) = 4p^2 + 6.5p + 7$	$W = \frac{4p+1}{p^2+p+1}, W_1 = \frac{5p+1}{6p^2+p+1}, W_2 = \frac{7p+1}{5p^2+p+1}$
4	$A(p) = 5p + 2.5;$ $B(p) = 5p^2 + 7.5p + 8$	$W = \frac{5p+1}{p^2+p+1}, W_1 = \frac{6p+1}{8p^2+p+1}, W_2 = \frac{8p+1}{6p^2+p+1}$
5	$A(p) = 6p + 1.5;$ $B(p) = 6p^2 + 4.5p + 5$	$W = \frac{6p+1}{p^2+p+1}, W_1 = \frac{5p+1}{6p^2+p+1}, W_2 = \frac{5p+1}{2p^2+p+1}$
6	$A(p) = 7p + 2.5;$ $B(p) = 7p^2 + 3.5p + 6$	$W = \frac{7p+1}{p^2+p+1}, W_1 = \frac{3p+1}{5p^2+p+1}, W_2 = \frac{6p+1}{2p^2+p+1}$
7	$A(p) = 8p + 3.5;$ $B(p) = 8p^2 + 7.5p + 9$	$W = \frac{8p+1}{p^2+p+1}, W_1 = \frac{5p+1}{8p^2+p+1}, W_2 = \frac{8p+1}{2p^2+p+1}$
8	$A(p) = 9p + 4.5;$ $B(p) = 9p^2 + 8.5p + 10$	$W = \frac{4p+1}{p^2+p+1}, W_1 = \frac{3p+1}{6p^2+p+1}, W_2 = \frac{7p+1}{3p^2+p+1}$
9	$A(p) = 3p + 4.5;$ $B(p) = 2p^2 + 4.5p + 6$	$W = \frac{5p+1}{p^2+p+1}, W_1 = \frac{4p+1}{6p^2+p+1}, W_2 = \frac{8p+1}{4p^2+p+1}$
10	$A(p) = 6p + 4.5;$ $B(p) = 4p^2 + 3.5p + 8$	$W = \frac{5p+1}{p^2+p+1}, W_1 = \frac{5p+1}{4p^2+p+1}, W_2 = \frac{7p+1}{3p^2+p+1}$
11	$A(p) = 5p + 2.5;$ $B(p) = 5p^2 + 3.5p + 6$	$W = \frac{4p+1}{p^2+p+1}, W_1 = \frac{5p+1}{6p^2+p+1}, W_2 = \frac{7p+1}{5p^2+p+1}$
12	$A(p) = 6p + 2.5;$ $B(p) = 6p^2 + 3.5p + 9$	$W = \frac{4p+1}{p^2+p+1}, W_1 = \frac{7p+1}{8p^2+p+1}, W_2 = \frac{6p+1}{4p^2+p+1}$
13	$A(p) = 7p + 1.5;$ $B(p) = 7p^2 + 4.5p + 5$	$W = \frac{6p+1}{p^2+p+1}, W_1 = \frac{4p+1}{6p^2+p+1}, W_2 = \frac{6p+1}{5p^2+p+1}$
14	$A(p) = 8p + 2.5;$ $B(p) = 8p^2 + 4.5p + 6$	$W = \frac{6p+1}{p^2+p+1}, W_1 = \frac{5p+1}{8p^2+p+1}, W_2 = \frac{4p+1}{2p^2+p+1}$
15	$A(p) = 5p + 3.5;$ $B(p) = 5p^2 + 4.5p + 8$	$W = \frac{8p+1}{p^2+p+1}, W_1 = \frac{6p+1}{4p^2+p+1}, W_2 = \frac{5p+1}{2p^2+p+1}$





Варианты к практическому занятию №1




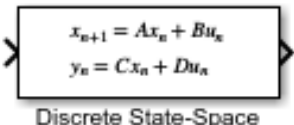
№ варианта	$A(p)$	$B(p)$
1	$A(p) = 2p + 1$	$B(p) = 2p^2 + 3p + 1$
2	$A(p) = 3p + 1$	$B(p) = 3p^2 + 2p + 1$
3	$A(p) = 4p + 1$	$B(p) = 4p^2 + 5p + 1$
4	$A(p) = 5p + 1$	$B(p) = 6p^2 + 4p + 1$
5	$A(p) = 6p + 1$	$B(p) = 5p^2 + 3p + 1$
6	$A(p) = 7p + 1$	$B(p) = 7p^2 + 5p + 1$
7	$A(p) = 8p + 1$	$B(p) = 8p^2 + 6p + 1$
8	$A(p) = 9p + 1$	$B(p) = 9p^2 + 5p + 1$
9	$A(p) = 2p + 1$	$B(p) = 3p^2 + 2p + 1$
10	$A(p) = 3p + 1$	$B(p) = 6p^2 + 4p + 1$
11	$A(p) = 4p + 1$	$B(p) = 5p^2 + 3p + 1$
12	$A(p) = 5p + 1$	$B(p) = 8p^2 + 6p + 1$
13	$A(p) = 6p + 1$	$B(p) = 9p^2 + 5p + 1$
14	$A(p) = 7p + 1$	$B(p) = 6p^2 + 4p + 1$
15	$A(p) = 8p + 1$	$B(p) = 4p^2 + 5p + 1$
Место для уравнения.		




Варианты к практическому занятию №2




№ варианта	Наименование блоков	Параметры настройки
1	Constant 	Constant value: 5; Sample time: inf; Output minimum: [1]; Output maximum: [5]
	Sum 	Output minimum: [1]; Output maximum: [1] и действием «+-»
	Gain 	Gain:5; Multiplication: Element-wise (K.*u)
	Saturation 	Upper limit: 4; Lower limit: -0.6




	Transfer Fcn 	Numerator coefficients: [2,9,3]; Denominator coefficients: [334,148,36,1]
	ПИД-регулятор 	Proportional (P): 3; Integral (I): 0.26, Derivative (D): 3
	Scope 	Number of input ports: 1; Sample time: 1; Input processing: Elements as channels (sample based); Maximize size of plots: off; Maximize size of plots: Manual; Time span: Auto; Time span overrun action: Wrap; Time units: None; Time display offset: 0; Time-axis labels: Bottom Displays Only
2	Sine Wave 	Amplitude: 10; Bias: 1; Frequency (rad/sec): 1; Phase (rad): 0; Sample time: 1
	Gain	Gain: 10; Multiplication: Element-wise (K.*u)
	Sum	Output minimum: [1]; Output maximum: [1] и действием «+-»
	Band-Limited White Noise 	Show port labels: FromPortIcon
	Transfer Fcn	Numerator coefficients: [2,4,5], Denominator coefficients: [345,146,26,1]
	ПИД-регулятор	Proportional (P): 2, Integral (I): 0.15, Derivative (D): 2
	Scope	Number of input ports: 2; Sample time: 1; Input processing: Elements as channels (sample based); Maximize size of plots: off; Maximize size of plots: Manual; Time span: Auto; Time span overrun action: Wrap; Time units: None; Time display offset: 0; Time-axis labels: Bottom Displays Only
3	Sine Wave	Amplitude: 15; Bias: 1; Frequency (rad/sec): 1; Phase (rad): 0; Sample time: 10
	Sum	Output minimum: [1], Output maximum: [1] и действием «+-»


	Terminator  Terminator	Исходные параметры
	Saturation	Upper limit: 5, Lower limit: -0.5
	Transfer Fcn	Numerator coefficients: [2,1]; Denominator coefficients: [2,3,1]
	ПИД-регулятор	Proportional (P): 3; Integral (I): 0.15; Derivative (D): 2
	Floating Scope  Floating Scope	Input processing: Elements as channels (sample based), Maximize axes: off, Axes scaling: Manual, Time span: Auto, Time span overrun action: Wrap, Time units: None, Time display offset: 0, Time-axis labels: Bottom Displays Only
4	Pulse Generator  Pulse Generator	Pulse type: Time based; Time: Use simulation time; Amplitude: 10; Period (secs): 1; Pulse Width (% of period): 5; Phase delay (secs): 1
	Sum	Output minimum: [1]; Output maximum: [1] и действием «+-»
	Gain	Gain: 5; Multiplication: Element-wise (K.*u)
	Derivative  Derivative	Coefficient c in the transfer function approximation $s/(c*s+1)$ used for linearization: Inf
	Transfer Fcn	Numerator coefficients: [3,1]; Denominator coefficients: [3,2,1]
	ПИД-регулятор	Proportional (P): 2; Integral (I): 0.15; Derivative (D): 3
	Scope	Number of input ports: 2; Sample time: 1; Input processing: Elements as channels (sample based); Maximize size of plots: off; Maximize size of plots: Manual; Time span: Auto; Time span overrun action: Wrap; Time units: None; Time display offset: 0; Time-axis labels: Bottom Displays Only
5	Step	Step time: 0.05, Final value: 1, Sample time: 0.1
	Sum	Output minimum: [1], Output maximum: [1] и действием «+-»
	Gain	Gain: 4; Multiplication: Element-wise (K.*u)
	Saturation	Upper limit: 4, Lower limit: -0.6
	Transfer Fcn	Numerator coefficients: [4,1], Denominator coefficients: [4,5,1]

	ПИД-регулятор	Proportional (P): 4, Integral (I): 0.35, Derivative (D): 4
	Display 	Numeric display format: long; Decimation: 1
6	Sine Wave	Amplitude: 10; Bias: 2; Frequency (rad/sec): 5; Phase (rad): 0; Sample time: 15
	Sum	Output minimum: [1], Output maximum: [1] и действием «+-»
	Saturation	Upper limit: 6, Lower limit: -0.1
	Transfer Fcn	Numerator coefficients: [5,1], Denominator coefficients: [6,4,1]
	ПИД-регулятор	Proportional (P): 2, Integral (I): 0.35, Derivative (D): 4
	Scope 	Number of input ports: 2; Sample time: 1; Input processing: Elements as channels (sample based); Maximize size of plots: off; Maximize size of plots: Manual; Time span: Auto; Time span overrun action: Wrap; Time units: None; Time display offset: 0; Time-axis labels: Bottom Displays Only
7	XY Graph 	x-min: 0; x-max: 100; y-min: -5; y-max: 200; Sample time: 1
	Step	Step time: 0.05, Final value: 1, Sample time: 0.1
	Sum	Output minimum: [1], Output maximum: [1] и действием «+-»
	Discrete State-Space 	$A = \begin{bmatrix} 0.9 & 0.15 \\ -0.7 & 0.58 \end{bmatrix}, B = \begin{bmatrix} 0.015 \\ 0.459 \end{bmatrix}, C = \begin{bmatrix} 0 & 1 \end{bmatrix}, D = \begin{bmatrix} 0 \end{bmatrix};$ Initial conditions: 0; Sample time (-1 for inherited): -1;
	Saturation	Upper limit: 4, Lower limit: -0.5
	Transfer Fcn	Numerator coefficients: [6,1], Denominator coefficients: [5,3,1]
	ПИД-регулятор	Proportional (P): 4, Integral (I): 0.35, Derivative (D): 2
	Scope	Number of input ports: 2; Sample time: 1; Input processing: Elements as channels (sample based); Maximize size of plots: off; Maximize size of plots: Manual; Time span: Auto; Time span overrun action: Wrap; Time units: None; Time display offset: 0; Time-axis labels: Bottom Displays Only

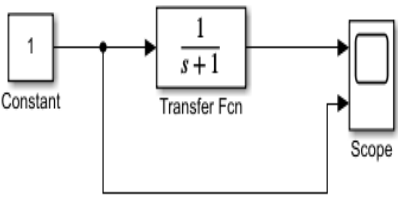
8	Sine Wave	Amplitude: 30; Bias: 5; Frequency (rad/sec): 10; Phase (rad): 0; Sample time: 10
	Sum	Output minimum: [1], Output maximum: [1] и действием «+-»
	Max 	Number of input: 2; Display option: bar
	Dead Zone 	Start of dead zone: -0.5; End of dead zone: 0.5
	Transfer Fcn	Numerator coefficients: [7,1], Denominator coefficients: [7,5,1]
	ПИД-регулятор	Proportional (P): 4, Integral (I): 0.45, Derivative (D): 4
	Scope	Number of input ports: 2; Sample time: 1; Input processing: Elements as channels (sample based); Maximize size of plots: off; Maximize size of plots: Manual; Time span: Auto; Time span overrun action: Wrap; Time units: None; Time display offset: 0; Time-axis labels: Bottom Displays Only
	XY Graph	x-min: 0; x-max: 10; y-min: 0; y-max: 100; Sample time: 0.5
9	Constant	Constant value: 3.6; Sample time: inf; Output minimum: [0]; Output maximum: [15]
	Product 	Number of input: 2; Multiplication: Element-wise(.*)
	Saturation	Upper limit: 4, Lower limit: -0.6
	Transfer Fcn	Numerator coefficients: [8,1], Denominator coefficients: [8,6,1]
	ПИД-регулятор	Proportional (P): 2, Integral (I): 0.35, Derivative (D): 2
	Display	Numeric display format: long; Decimation: 1
	Step	Step time: 0.05, Final value: 1, Sample time: 0.1
10	Pulse Generator	Pulse type: Time based; Time: Use simulation time; Amplitude: 10; Period (secs): 1; Pulse Width (% of period): 5; Phase delay (secs): 1
	Sum	Output minimum: [1], Output maximum: [1] и действием «+-»

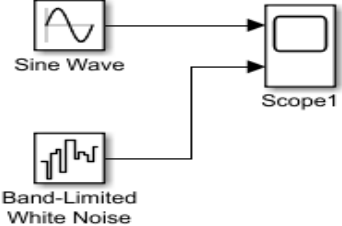
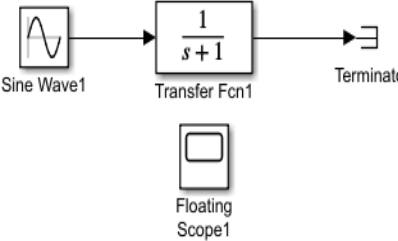
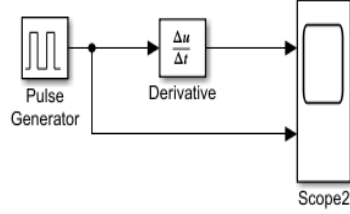
	Rate Limiter  Rate Limiter	Rising slew rate: 2; Falling slew rate: -1; Sample time mode: inherited; Initial condition: 0
	Saturation	Upper limit: 5, Lower limit: -0.1
	Transfer Fcn	Numerator coefficients: [9,1], Denominator coefficients: [9,5,1]
	ПИД-регулятор	Proportional (P): 3, Integral (I): 0.45, Derivative (D): 3
	Scope	Number of input ports: 2; Sample time: 1; Input processing: Elements as channels (sample based); Maximize size of plots: off; Maximize size of plots: Manual; Time span: Auto; Time span overrun action: Wrap; Time units: None; Time display offset: 0; Time-axis labels: Bottom Displays Only
11	Sine Wave	Amplitude: 10; Bias: 2; Frequency (rad/sec): 5; Phase (rad): 0; Sample time: 30
	Sum	Output minimum: [1], Output maximum: [1] и действием «+-»
	Quantizer  Quantizer	Quantization interval: 0.6
	Saturation	Upper limit: 6, Lower limit: -0.4
	Transfer Fcn	Numerator coefficients: [2,1], Denominator coefficients: [3,2,1]
	ПИД-регулятор	Proportional (P): 2, Integral (I): 0.45, Derivative (D): 3
	Scope	Number of input ports: 2, Sample time: 1, Input processing: Elements as channels (sample based), Maximize size of plots: off, Maximize size of plots: Manual, Time span: Auto, Time span overrun action: Wrap, Time units: None, Time display offset: 0, Time-axis labels: Bottom Displays Only
12	Sine Wave	Amplitude: 15; Bias: 2; Frequency (rad/sec): 5; Phase (rad): 0; Sample time: 45
	Sum	Output minimum: [1], Output maximum: [1] и действием «+-»
	Coulomb & Viscous Friction  Coulomb & Viscous Friction	Show port labels: FromPortIcon
	Max	Number of input: 2; Display option: bar

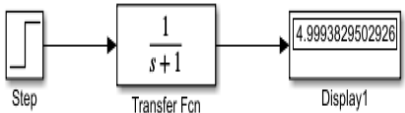
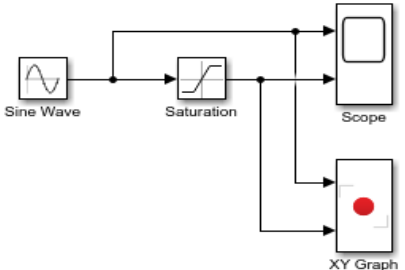
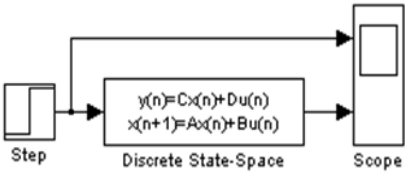
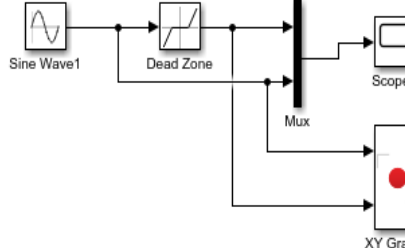
	Transfer Fcn	Numerator coefficients: [3,1], Denominator coefficients: [6,4,1]
	ПИД-регулятор	Proportional (P): 3, Integral (I): 0.45, Derivative (D): 2
	Scope	Number of input ports: 2, Sample time: 1, Input processing: Elements as channels (sample based), Maximize size of plots: off, Maximize size of plots: Manual, Time span: Auto, Time span overrun action: Wrap, Time units: None, Time display offset: 0, Time-axis labels: Bottom Displays Only
	XY Graph	x-min: 0; x-max: 15; y-min: -50; y-max: 50; Sample time: 0.1
13	Sine Wave	Amplitude: 40; Bias: 1; Frequency (rad/sec): 1; Phase (rad): 0; Sample time: 10
	Ramp 	Show port labels: FromPortIcon
	Product	Number of input: 2; Multiplication: Element-wise(.*)
	Backlash 	Deadband width: 1, Initial output: 0
	Max	Number of input: 2; Display option: bar
	ПИД-регулятор	Proportional (P): 4, Integral (I): 0.35, Derivative (D): 2
	Scope	Number of axes: 2, Sample time: 1, Input processing: Elements as channels (sample based), Maximize size of plots: off, Maximize size of plots: Manual, Time span: Auto, Time span overrun action: Wrap, Time units: None, Time display offset: 0, Time-axis labels: Bottom Displays Only
	XY Graph	x-min: 0; x-max: 10; y-min: 0; y-max: 50; Sample time: 0.5
14	Constant	Constant value: [-0.5 1;2 5] ; Sample time: inf; Output minimum: [-5]; Output maximum: [5]
	Sum	Output minimum: [1], Output maximum: [1] и действием «+-»
	Add 	Icon shape: rectangular; List of signs: ++; Accumulator data type: Inherit: Inherit via internal rule (default); Output minimum: [1]; Output maximum:

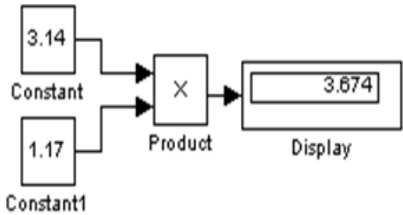
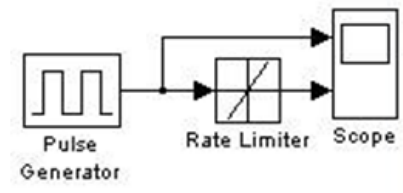
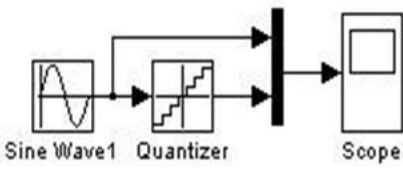
		[10]; Output data type: Inherit: Inherit via internal rule (default)
	Saturation	Upper limit: 5, Lower limit: -0.1
	Transfer Fcn	Numerator coefficients: [5,1], Denominator coefficients: [8,6,1]
	ПИД-регулятор	Proportional (P): 3, Integral (I): 0.45, Derivative (D): 2
	Display	Numeric display format: long; Decimation: 1
15	Constant	Constant value: [1.5 3.6 4.5 5.8]; Sample time: inf; Output minimum: [-6]; Output maximum: [6]
	Sum of Elements 	Icon shape: rectangular; List of signs: +; Sum over: All dimensions; Accumulator data type: Inherit: Inherit via internal rule (default); Output minimum: [1]; Output maximum: [10]; Output data type: Inherit: Inherit via internal rule (default)
	Gain	Gain: 5; Multiplication: Element-wise (K.*u)
	Saturation	Upper limit: 10, Lower limit: -0.5
	Transfer Fcn	Numerator coefficients: [6,1]; Denominator coefficients: [9,5,1]
	ПИД-регулятор	Proportional (P): 2; Integral (I): 0.45; Derivative (D): 2
	Display	Numeric display format: long; Decimation: 1

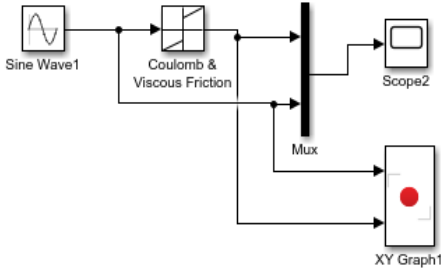
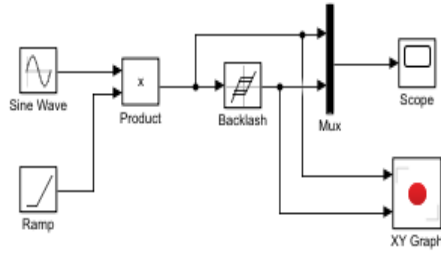
Варианты к практическому занятию №3

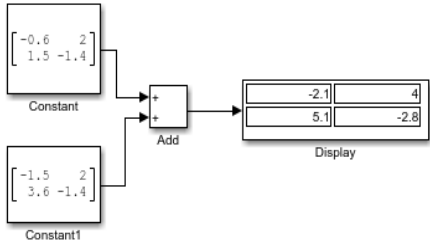
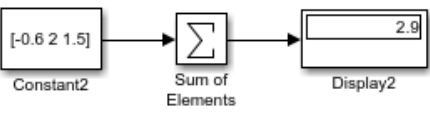
№ варианта	Структурная модель	Параметры настройки
1		<p>Constant: Constant value: 5; Sample time: inf; Output minimum: [1]; Output maximum: [5];</p> <p>Transfer Fcn: Numerator coefficients: [4,1], Denominator coefficients: [4,5,1];</p> <p>Scope: Number of input ports: 2; Sample time: 1; Input processing: Elements as channels (sample based); Maximize size of plots: off; Maximize size of plots: Manual; Time span: Auto; Time span overrun action: Wrap; Time units: None; Time display offset: 0; Time-axis labels: Bottom Displays Only</p>

2		<p>Sine Wave: Amplitude: 10; Bias: 1; Frequency (rad/sec): 1; Phase (rad): 0; Sample time: 1;</p> <p>Band-Limited White Noise: Show port labels: FromPortIcon;</p> <p>Scope1: Number of input ports: 2; Sample time: 1; Input processing: Elements as channels (sample based); Maximize size of plots: off; Maximize size of plots: Manual; Time span: Auto; Time span overrun action: Wrap; Time units: None; Time display offset: 0; Time-axis labels: Bottom Displays Only</p>
3		<p>Sine Wave1: Amplitude: 15; Bias: 1; Frequency (rad/sec): 1; Phase (rad): 0; Sample time: 10;</p> <p>Transfer Fcn1: Numerator coefficients: [2,1]; Denominator coefficients: [2,3,1];</p> <p>Terminator: Исходные параметры;</p> <p>Floating Scope1: Input processing: Elements as channels (sample based), Maximize axes: off, Axes scaling: Manual, Time span: Auto, Time span overrun action: Wrap, Time units: None, Time display offset: 0, Time-axis labels: Bottom Displays Only</p>
4		<p>Pulse Generator: Pulse type: Time based; Time: Use simulation time; Amplitude: 10; Period (secs): 1; Pulse Width (% of period): 5; Phase delay (secs): 1;</p> <p>Derivation: Coefficient c in the transfer function approximation $s/(c*s+1)$ used for linearization: Inf;</p> <p>Scope2: Number of input ports: 2; Sample time: 1; Input processing: Elements as channels (sample based); Maximize size of plots: off; Maximize size of plots: Manual; Time span: Auto; Time span overrun action: Wrap; Time units: None; Time display offset: 0; Time-axis labels: Bottom Displays Only</p>


5		<p>Step: Step time: 0.05, Final value: 1, Sample time: 0.1;</p> <p>Transfer Fcn: Numerator coefficients: [4,1], Denominator coefficients: [4,5,1];</p> <p>Display1: Numeric display format: long; Decimation: 1</p>
6		<p>Sine Wave: Amplitude: 10; Bias: 2; Frequency (rad/sec): 5; Phase (rad): 0; Sample time: 15;</p> <p>Saturation: Upper limit: 6, Lower limit: -0.1;</p> <p>Scope: Number of input ports: 2; Sample time: 1; Input processing: Elements as channels (sample based); Maximize size of plots: off; Maximize size of plots: Manual; Time span: Auto; Time span overrun action: Wrap; Time units: None; Time display offset: 0; Time-axis labels: Bottom Displays Only;</p> <p>XY Graph: x-min: 0; x-max: 100; y-min: -5; y-max: 200; Sample time: 1</p>
7		<p>Step: Step time: 0.05; Final value: 10; Sample time: 0.1;</p> <p>Discrete State-Space:</p> $A = \begin{bmatrix} 0.9899 & 0.1599 \\ -0.7499 & 0.5899 \end{bmatrix}, B = \begin{bmatrix} 0.0159 \\ 0.4599 \end{bmatrix}, C = \begin{bmatrix} 0 & 1 \end{bmatrix}, D = \begin{bmatrix} 0 \end{bmatrix};$ <p>Initial conditions: 0;</p> <p>Sample time (-1 for inherited): -1;</p> <p>Scope: Number of input ports: 2; Sample time: 1; Input processing: Elements as channels (sample based); Maximize size of plots: off; Maximize size of plots: Manual; Time span: Auto; Time span overrun action: Wrap; Time units: None; Time display offset: 0; Time-axis labels: Bottom Displays Only</p>
8		<p>Sine Wave1: Amplitude: 30; Bias: 5; Frequency (rad/sec): 10; Phase (rad): 0; Sample time: 10;</p> <p>Dead Zone: Start of dead zone: -0.5; End of dead zone: 0.5;</p>

		<p>Mux: Number of input: 2; Display option: bar;</p> <p>Scope2: Number of input ports: 1; Sample time: 1; Input processing: Elements as channels (sample based); Maximize size of plots: off; Maximize size of plots: Manual; Time span: Auto; Time span overrun action: Wrap; Time units: None; Time display offset: 0; Time-axis labels: Bottom Displays Only;</p> <p>XY Graph: x-min: 0; x-max: 10; y-min: 0; y-max: 100; Sample time: 0.5</p>
9		<p>Constant: Constant value: 3.14; Sample time: inf; Output minimum: [0]; Output maximum: [15];</p> <p>Constant1: Constant value: 1.17; Sample time: inf; Output minimum: [0]; Output maximum: [15];</p> <p>Product: Number of input: 2; Multiplication: Element-wise(.*);</p> <p>Display: Numeric display format: long; Decimation: 1</p>
10		<p>Pulse Generator: Pulse type: Time based; Time: Use simulation time; Amplitude: 10; Period (secs): 1; Pulse Width (% of period): 5; Phase delay (secs): 1;</p> <p>Rate Limiter: Rising slew rate: 2; Falling slew rate: -1; Sample time mode: inherited; Initial condition: 0;</p> <p>Scope: Number of input ports: 2; Sample time: 1; Input processing: Elements as channels (sample based); Maximize size of plots: off; Maximize size of plots: Manual; Time span: Auto; Time span overrun action: Wrap; Time units: None; Time display offset: 0; Time-axis labels: Bottom Displays Only</p>
11		<p>Sine Wave1: Amplitude: 10; Bias: 2; Frequency (rad/sec): 5; Phase (rad): 0; Sample time: 30;</p> <p>Quantizer: Quantization interval: 0.6;</p> <p>Mux: Number of input: 2; Display option: bar;</p>

		<p>Scope: Number of input ports: 1, Sample time: 1, Input processing: Elements as channels (sample based), Maximize size of plots: off, Maximize size of plots: Manual, Time span: Auto, Time span overrun action: Wrap, Time units: None, Time display offset: 0, Time-axis labels: Bottom Displays Only</p>
12		<p>Sine Wave1: Amplitude: 15; Bias: 2; Frequency (rad/sec): 5; Phase (rad): 0; Sample time: 45;</p> <p>Coulomb & Viscous Friction: Show port labels: FromPortIcon;</p> <p>Mux: Number of input: 2; Display option: bar;</p> <p>Scope2: Number of input ports: 1, Sample time: 1, Input processing: Elements as channels (sample based), Maximize size of plots: off, Maximize size of plots: Manual, Time span: Auto, Time span overrun action: Wrap, Time units: None, Time display offset: 0, Time-axis labels: Bottom Displays Only;</p> <p>XY Graph: x-min: 0; x-max: 15; y-min: -50; y-max: 50; Sample time: 0.1</p>
13		<p>Sine Wave: Amplitude: 40; Bias: 1; Frequency (rad/sec): 1; Phase (rad): 0; Sample time: 10;</p> <p>Ramp: Show port labels: FromPortIcon;</p> <p>Product: Number of input: 2; Multiplication: Element-wise(.*);</p> <p>Backlash: Deadband width: 1, Initial output: 0;</p> <p>Mux: Number of input: 2; Display option: bar;</p> <p>Scope: Number of input ports: 1, Sample time: 1, Input processing: Elements as channels (sample based), Maximize size of plots: off, Maximize size of plots: Manual, Time span: Auto, Time span overrun action: Wrap, Time units: None, Time display offset: 0, Time-axis labels: Bottom Displays Only;</p>

		XY Graph: x-min: 0; x-max: 10; y-min: 0; y-max: 50; Sample time: 0.5
14		<p>Constant: Constant value: [-0.5 1; 2 5]; Sample time: inf; Output minimum: [-5]; Output maximum: [5];</p> <p>Constant1: Constant value: [-0.6 2; 2 6]; Sample time: inf; Output minimum: [-5]; Output maximum: [5];</p> <p>Add: Constant value: [-0.5 1; 2 5]; Sample time: inf; Output minimum: [-5]; Output maximum: [5];</p> <p>Display: Numeric display format: long; Decimation: 1</p>
15		<p>Constant2: Constant value: [1.5 3.6 4.5 5.8]; Sample time: inf; Output minimum: [-6]; Output maximum: [6];</p> <p>Sum of Elements: Icon shape: rectangular; List of signs: +; Sum over: All dimensions; Accumulator data type: Inherit: Inherit via internal rule (default); Output minimum: [1]; Output maximum: [10]; Output data type: Inherit: Inherit via internal rule (default);</p> <p>Display2: Numeric display format: long; Decimation: 1</p>

Варианты к практическому занятию №14

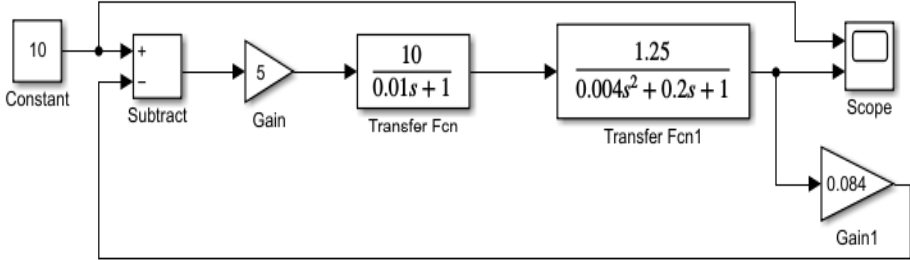
№ варианта	Структурная модель	Характеристики
	<div style="text-align: center;">  </div> <p>Параметры настройки Stop Time: 300 с</p> <p>Step: Step time: 0, Initial value: 0, Final value: 1, Sample time: 0; Sum: действие « +-»;</p> <p>Saturation: Upper limit: 5, Lower limit: -0.5; Transfer Fcn: Numerator coefficients: [1 8 2], Denominator coefficients: [336 145 25 1]; ПИД-регулятор: Controller: PID, Proportional (P): 0.5, Integral (I): 0.01, Derivative (D): 1, Filter coefficient (N): 100; Scope: Number of input ports: 1, Sample time: 1, Input processing: Elements as channels (sample based), Maximize size of plots: off, Maximize size of plots: Manual, Time span: Auto, Time span overrun action: Wrap, Time units: None, Time display offset: 0, Time-axis labels: Bottom Displays Only</p>	<p>Step, Nyquist, Zero Map</p> <p>Bode, Pole-</p>

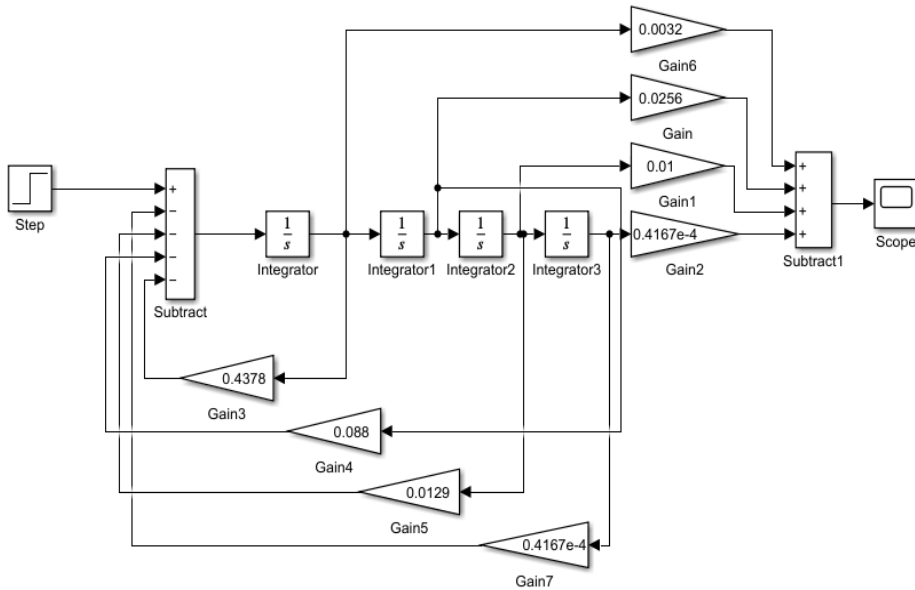
2	<div data-bbox="548 272 1444 547" data-label="Diagram"> </div> <p>Параметры настройки Stop Time: 30 с</p> <p>Step: Step time: 0, Initial value: 0, Final value: 1, Sample time: 0; Sum: действие « +-»»; Gain: Gain: 1, Multiplication: Element-wise (K.*u); Transfer Fcn: Numerator coefficients: [1], Denominator coefficients: [1 0.6 1]; Transfer Fcn1: Numerator coefficients: [1], Denominator coefficients: [0.1 1]; Scope: Number of input ports: 1, Sample time: 1, Input processing: Elements as channels (sample based), Maximize size of plots: off, Maximize size of plots: Manual, Time span: Auto, Time span overrun action: Wrap, Time units: None, Time display offset: 0, Time-axis labels: Bottom Displays Only</p>	<p>Step, Nyquist, Zero Map</p> <p>Bode, Pole-</p>
3	<div data-bbox="627 1061 1400 1241" data-label="Diagram"> </div> <p>Параметры настройки Stop Time: 100 с</p>	<p>Step, Bode, Nyquist, Zero Map</p> <p>Impulse, Nyquist, Pole-</p>

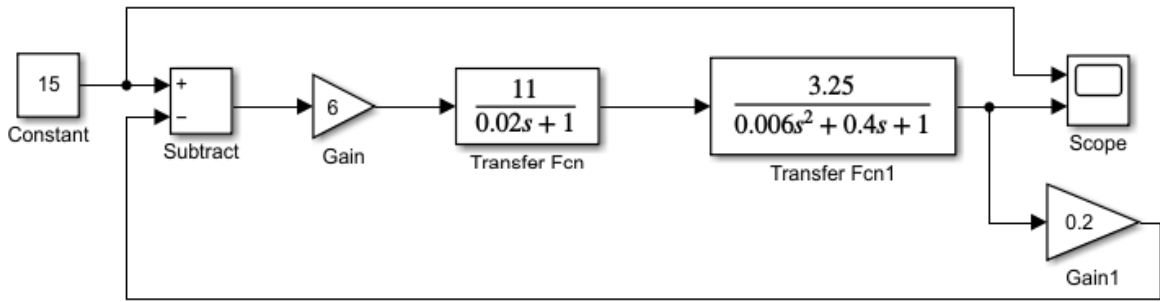
	<p>Step: Step time: 2, Initial value: 0, Final value: 5, Sample time: 0; Mux: Number of input: 2;</p> <p>Display option: bar;</p> <p>State-Space:</p> $A = \begin{bmatrix} 0 & 1 & 0 \\ -5 & -2 & 0 \\ 0 & 0 & -1 \end{bmatrix}, B = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, C = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, D = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix};$ <p>Initial conditions: [0; 0; 5];</p> <p>Demux: Number of outputs: 3; Display option: bar; Scope: Number of input ports: 3, Sample time: 1, Input processing: Elements as channels (sample based), Maximize size of plots: off, Maximize size of plots: Manual, Time span: Auto, Time span overrun action: Wrap, Time units: None, Time display offset: 0, Time-axis labels: Bottom Displays Only</p>	
4	<p>Параметры настройки</p> <p>Stop Time: 300 c</p>	<p>Step, Impulse, Bode, Nyquist, Pole-Zero Map</p>

	<p> Step: Step time: 0, Initial value: 0, Final value: 0, Sample time: 0; Step1: Step time: 0, Initial value, 0, Final value: 10, Sample time: 0; Sum: действие « +-»»; Sum1: действие «++ »»; ПИД-регулятор: Controller: PD, Proportional (P): 0.5, Derivative (D): 1, Filter coefficient (N): 10; Transfer Fcn: Numerator coefficients: [1], Denominator coefficients: [3 1]; Transfer Fcn1: Numerator coefficients: [0.06], Denominator coefficients: [12 1]; Transfer Fcn2: Numerator coefficients: [1], Denominator coefficients: [5 1]; Scope: Number of input ports: 1, Sample time: 1, Input processing: Elements as channels (sample based), Maximize size of plots: off, Maximize size of plots: Manual, Time span: Auto, Time span overrun action: Wrap, Time units: None, Time display offset: 0, Time-axis labels: Bottom Displays Only; Scope1: Number of input ports: 1, Sample time: 1, Input processing: Elements as channels (sample based), Maximize size of plots: off, Maximize size of plots: Manual, Time span: Auto, Time span overrun action: Wrap, Time units: None, Time display offset: 0, Time-axis labels: Bottom Displays Only </p>	
5		<p> Step, Impulse, Bode, Nyquist, Pole-Zero Map </p>

	<p>Параметры настройки Stop Time: 100 с</p> <p>Step: Step time: 0, Initial value: 0, Final value: 5, Sample time: 0; Subtract: Icon shape: rectangular, List of signs: «+---»;</p> <p>Subtract1: Icon shape: rectangular, List of signs: «++++»;</p> <p>Integrator: External reset: none, Initial condition source: internal, Initial condition: 0; Integrator1: External reset: none, Initial condition source: internal, Initial condition: 0; Integrator2: External reset: none, Initial condition source: internal, Initial condition: 0; Gain: Gain: 0.0071, Multiplication: Element-wise (K.*u); Gain1: Gain: 0.0555, Multiplication: Element-wise (K.*u); Gain2: Gain: 0.0174, Multiplication: Element-wise (K.*u); Gain3: Gain: 0.4417, Multiplication: Element-wise (K.*u); Gain4: Gain: 0,118, Multiplication: Element-wise (K.*u); Gain5: Gain: 0.0204, Multiplication: Element-wise (K.*u); Scope: Number of input ports: 1, Sample time: 1, Input processing: Elements as channels (sample based), Maximize size of plots: off, Maximize size of plots: Manual, Time span: Auto, Time span overrun action: Wrap, Time units: None, Time display offset: 0, Time-axis labels: Bottom Displays Only</p>	
6		<p>Step, Impulse, Bode, Nyquist, Pole-Zero Map</p>

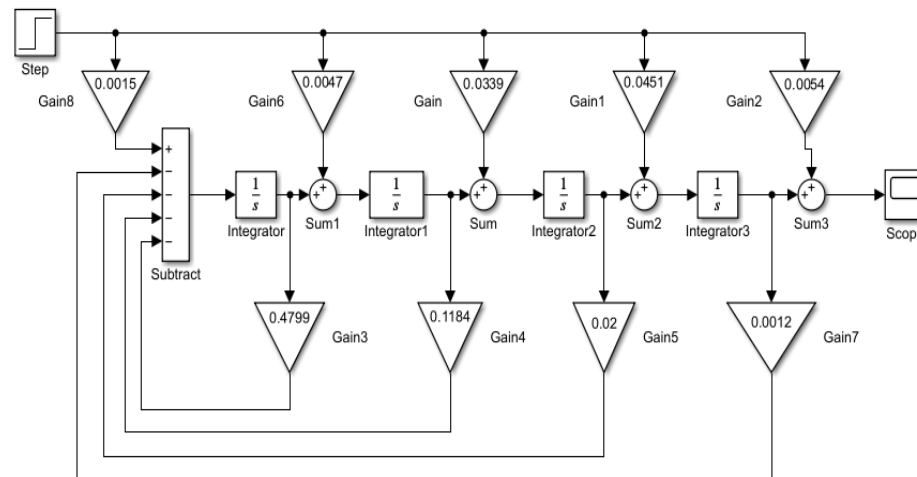
	<p>Параметры настройки Stop Time: 200 с</p> <p>Step: Step time: 1, Initial value: 0, Final value: 1, Sample time: 0; Sum: действие « +-»»; Sum1: действие « +-»»; Gain: Gain: 0.6, Multiplication: Element-wise (K.*u); Gain1: Gain: 2, Multiplication: Element-wise (K.*u); Gain2: Gain: 0.4, Multiplication: Element-wise (K.*u); Integrator: External reset: none, Initial condition source: internal, Initial condition: 0; Transfer Fcn: Numerator coefficients: [10], Denominator coefficients: [0.01 0.2 1]; Transfer Fcn1: Numerator coefficients: [0.8], Denominator coefficients: [5 1]; Transfer Fcn2: Numerator coefficients: [1], Denominator coefficients: [0.3 1]; Scope: Number of input ports: 1, Sample time: 1, Input processing: Elements as channels (sample based), Maximize size of plots: off, Maximize size of plots: Manual, Time span: Auto, Time span overrun action: Wrap, Time units: None, Time display offset: 0, Time-axis labels: Bottom Displays Only</p>	
7	 <p>Параметры настройки Stop Time: 1.5 с</p> <p>Constant: Constant value: 10, Sample time: inf; Gain: Gain: 5, Multiplication: Element-wise (K.*u); Gain1: Gain: 0.084, Multiplication: Element-wise (K.*u); Subtract: Icon</p>	<p>Step, Impulse, Bode, Nyquist, Pole-Zero Map</p>

	<p>shape: rectangular, List of signs: «+-»; Transfer Fcn: Numerator coefficients: [10], Denominator coefficients: [0.01 1]; Transfer Fcn1: Numerator coefficients: [1.25], Denominator coefficients: [0.004 0.2 1]; Scope: Number of input ports: 2, Sample time: 1, Input processing: Elements as channels (sample based), Maximize size of plots: off, Maximize size of plots: Manual, Time span: Auto, Time span overrun action: Wrap, Time units: None, Time display offset: 0, Time-axis labels: Bottom Displays Only</p>	
8	 <p>Параметры настройки Stop Time: 100 с</p> <p>Step: Step time: 0, Initial value: 0, Final value: 5, Sample time: 0; Subtract: Icon shape: rectangular, List of signs: «+----»; Subtract1: Icon shape: rectangular, List of signs: «++++»;</p>	<p>Step, Impulse, Bode, Nyquist, Pole-Zero Map</p>

	<p>Integrator: External reset: none, Initial condition source: internal, Initial condition: 0; Integrator1: External reset: none, Initial condition source: internal, Initial condition: 0; Integrator2: External reset: none, Initial condition source: internal, Initial condition: 0; Integrator3: External reset: none, Initial condition source: internal, Initial condition: 0; Gain: Gain: 0.0256, Multiplication: Element-wise (K.*u); Gain1: Gain: 0.01, Multiplication: Element-wise (K.*u); Gain2: Gain: 0.4176e-4, Multiplication: Element-wise (K.*u); Gain3: Gain: 0.4378, Multiplication: Element-wise (K.*u); Gain4: Gain: 0.088, Multiplication: Element-wise (K.*u); Gain5: Gain: 0.0129, Multiplication: Element-wise (K.*u); Gain6: Gain: 0.0032, Multiplication: Element-wise (K.*u); Gain7: Gain: 0.4167e-4, Multiplication: Element-wise (K.*u); Scope: Number of input ports: 1, Sample time: 1, Input processing: Elements as channels (sample based), Maximize size of plots: off, Maximize size of plots: Manual, Time span: Auto, Time span overrun action: Wrap, Time units: None, Time display offset: 0, Time-axis labels: Bottom Displays Only</p>	
9	 <p>Параметры настройки Stop Time: 10 c</p>	<p>Step, Nyquist, Zero Map</p> <p>Bode, Pole-</p>

Constant: Constant value: 15, Sample time: inf; **Gain:** Gain: 6, **Multiplication:** Element-wise (K.*u); **Gain1:** Gain: 0.2, **Multiplication:** Element-wise (K.*u); **Subtract:** Icon shape: rectangular, **List of signs:** «+-»; **Transfer Fcn:** Numerator coefficients: [11], Denominator coefficients: [0.02 1]; **Transfer Fcn1:** Numerator coefficients: [3.25], Denominator coefficients: [0.006 0.4 1]; **Scope:** Number of input ports: 2, Sample time: 1, **Input processing:** Elements as channels (sample based), **Maximize size of plots:** off, **Maximize size of plots:** Manual, **Time span:** Auto, **Time span overrun action:** Wrap, **Time units:** None, **Time display offset:** 0, **Time-axis labels:** Bottom Displays Only

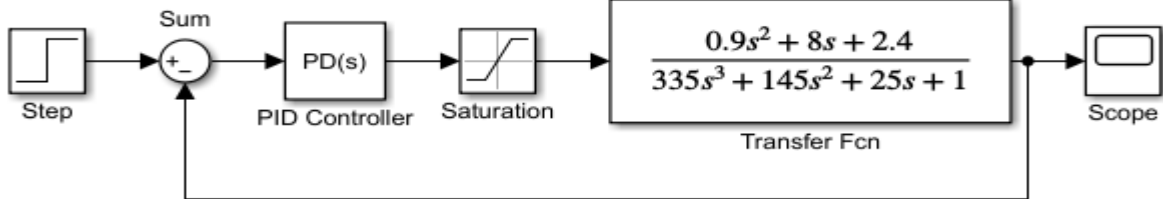
10

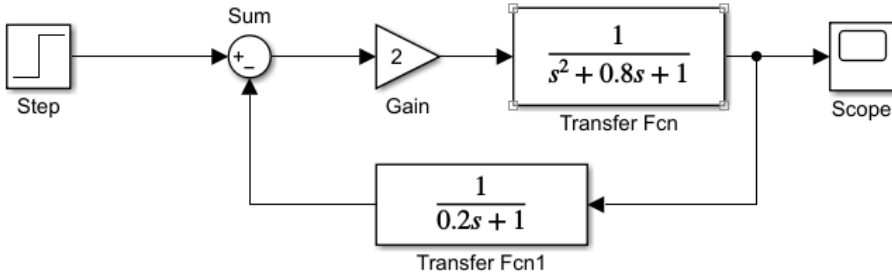


Параметры настройки
Stop Time: 150 c

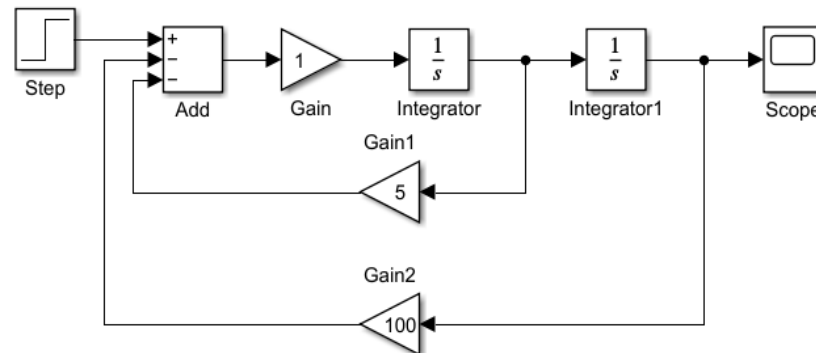
Step, Impulse, Bode, Nyquist, Pole-Zero Map

	<p>Step: Step time: 0, Initial value: 0, Final value: 5, Sample time: 0; Gain: Gain: 0.0339, Multiplication: Element-wise (K.*u); Gain1: Gain: 0.0451, Multiplication: Element-wise (K.*u); Gain2: Gain: 0.0054, Multiplication: Element-wise (K.*u); Gain3: Gain: 0.4799, Multiplication: Element-wise (K.*u); Gain4: Gain: 0.1184, Multiplication: Element-wise (K.*u); Gain5: Gain: 0.02, Multiplication: Element-wise (K.*u); Gain6: Gain: 0.0047, Multiplication: Element-wise (K.*u); Gain7: Gain: 0.0012, Multiplication: Element-wise (K.*u); Gain8: Gain: 0.0015, Multiplication: Element-wise (K.*u); Subtract: Icon shape: rectangular, List of signs: «+---»; Integrator: External reset: none, Initial condition source: internal, Initial condition: 0; Integrator1: External reset: none, Initial condition source: internal, Initial condition: 0; Integrator2: External reset: none, Initial condition source: internal, Initial condition: 0; Integrator3: External reset: none, Initial condition source: internal, Initial condition: 0; Sum: действие «++ »; Sum1: действие «++ »; Sum2: действие «++ »; Sum3: действие «++ »; Scope: Number of input ports: 1, Sample time: 1, Input processing: Elements as channels (sample based), Maximize size of plots: off, Maximize size of plots: Manual, Time span: Auto, Time span overrun action: Wrap, Time units: None, Time display offset: 0, Time-axis labels: Bottom Displays Only</p>	
11		<p>Step, Impulse, Bode, Nyquist, Pole-Zero Map</p>

	<p>Параметры настройки Stop Time: 300 с</p> <p>Step: Step time: 0, Initial value: 0, Final value: 0, Sample time: 0; Step1: Step time: 0, Initial value: 0, Final value: 10, Sample time: 0; Sum: действие « +-»»; Sum1: действие «++ »»; ПИД-регулятор: Controller: P, Proportional (P): 0.5; Transfer Fcn: Numerator coefficients: [1], Denominator coefficients: [4 1]; Transfer Fcn1: Numerator coefficients: [0.06], Denominator coefficients: [15 1]; Transfer Fcn2: Numerator coefficients: [1], Denominator coefficients: [6 1]; Scope: Number of axes: 1, Sample time: 1, Input processing: Elements as channels (sample based), Maximize size of plots: off, Maximize size of plots: Manual, Time span: Auto, Time span overrun action: Wrap, Time units: None, Time display offset: 0, Time-axis labels: Bottom Displays Only; Scope1: Number of input ports: 1, Sample time: 1, Input processing: Elements as channels (sample based), Maximize size of plots: off, Maximize size of plots: Manual, Time span: Auto, Time span overrun action: Wrap, Time units: None, Time display offset: 0, Time-axis labels: Bottom Displays Only</p>	
12	 <p>Параметры настройки Stop Time: 200 с</p>	<p>Step, Impulse, Bode, Nyquist, Pole-Zero Map</p>

	<p>Step: Step time: 0, Initial value: 0, Final value: 1, Sample time: 0; Sum: действие « +-»; ПИД-регулятор: Controller: PD, Proportional (P): 2, Derivative (D): 1, Filter coefficient (N): 10; Saturation: Upper limit: 5, Lower limit: -0.5; Transfer Fcn: Numerator coefficients: [0.9 8 2.4], Denominator coefficients: [335 146 25 1]; Scope: Number of input ports: 1, Sample time: 1, Input processing: Elements as channels (sample based), Maximize size of plots: off, Maximize size of plots: Manual, Time span: Auto, Time span overrun action: Wrap, Time units: None, Time display offset: 0, Time-axis labels: Bottom Displays Only</p>	
13	 <p>Параметры настройки Stop Time: 30 с</p> <p>Step: Step time: 0, Initial value: 0, Final value: 1, Sample time: 0; Sum: действие « +-»; Gain: Gain: 2, Multiplication: Element-wise (K.*u); Transfer Fcn: Numerator coefficients: [1], Denominator coefficients: [1 0.8 1]; Transfer Fcn1: Numerator coefficients: [1], Denominator coefficients: [0.2 1]; Scope: Number of input ports: 1, Sample time: 1, Input processing: Elements as channels (sample based), Maximize size of plots: off, Maximize size of plots: Manual, Time span: Auto, Time span overrun action: Wrap, Time units: None, Time display offset: 0, Time-axis labels: Bottom Displays Only</p>	<p>Step, Impulse, Bode, Nyquist, Pole-Zero Map</p>

14



Параметры настройки

Stop Time: 10 с

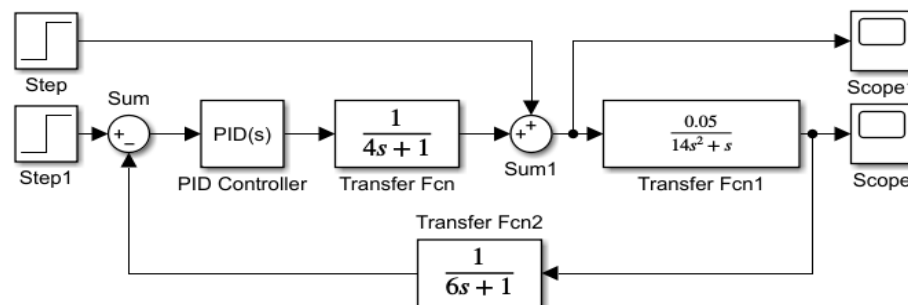
Step: Step time: 0, Initial value: 0, Final value: 5, Sample time: 0; **Add:** Icon shape: rectangular, List of signs «+--»;

Gain: Gain: 1, Multiplication: Element-wise (K.*u); **Gain1:** Gain: 5, Multiplication: Element-wise (K.*u); **Gain2:** Gain: 100, Multiplication: Element-wise (K.*u);

Integrator: External reset: none, Initial condition source: internal, Initial condition: 0; **Integrator1:** External reset: none, Initial condition source: internal, Initial condition: 0; **Scope:** Number of input ports: 1, Sample time: 1, Input processing: Elements as channels (sample based), Maximize size of plots: off, Maximize size of plots: Manual, Time span: Auto, Time span overrun action: Wrap, Time units: None, Time display offset: 0, Time-axis labels: Bottom Displays Only

Step, Impulse, Bode, Nyquist, Pole-Zero Map

15



Параметры настройки

Stop Time: 300 с

Step: Step time: 0, Initial value: 0, Final value: 0, Sample time: 0; **Step1:** Step time: 0, Initial value: 0, Final value: 10, Sample time: 0; **Sum:** действие «|+-»»; **Sum1:** действие «++|»; **ПИД-регулятор:** Controller: PID, Proportional (P): 0.5, Integral (I): 0.005, Derivative (D): 0.5, Filter coefficient (N): 10; **Transfer Fcn:** Numerator coefficients: [1], Denominator coefficients: [4 1]; **Transfer Fcn1:** Numerator coefficients: [0.05], Denominator coefficients: [14 1]; **Transfer Fcn2:** Numerator coefficients: [1], Denominator coefficients: [6 1]; **Scope:** Number of input ports: 1, Sample time: 1, Input processing: Elements as channels (sample based), Maximize size of plots: off, Maximize size of plots: Manual, Time span: Auto, Time span overrun action: Wrap, Time units: None, Time display offset: 0, Time-axis labels: Bottom Displays Only; **Scope1:** Number of axes: 1, Sample time: 1, Input processing: Elements as channels (sample based), Maximize size of plots: off, Maximize size of plots: Manual, Time span: Auto, Time span overrun action: Wrap, Time units: None, Time display offset: 0, Time-axis labels: Bottom Displays Only

Step, Impulse, Bode, Nyquist, Pole-Zero Map