

Министерство образования и науки Российской Федерации
Ивановский государственный химико-технологический университет

Э.Г. ГАЛИАСКАРОВ, Т.В. ЛАБУТИНА

МОДЕЛИРОВАНИЕ СИСТЕМ

ЛАБОРАТОРНЫЙ ПРАКТИКУМ

Иваново 2010

УДК 681.3

Галиаскаров Э.Г., Лабутина Т.В. Моделирование систем: лабораторный практикум/ Иван. гос. хим.-технол. ун-т.- Иваново, 2010.- 128с.

Лабораторный практикум ставит себе целью познакомить студентов с основными положениями теории моделирования систем, привить навыки использования технологии имитационного моделирования систем и процессов. Предназначен для студентов специальности «Информационные системы и технологии», занимающихся по дисциплинам «Функциональные математические модели» и «Моделирование систем».

Печатается по решению редакционно-издательского совета Ивановского государственного химико-технологического университета

Рецензенты:

доктор технических наук, профессор В.Е. Мизонов (Ивановский государственный энергетический университет);

кандидат технических наук, доцент П.В. Грименицкий (Ивановский государственный химико-технологический университет)

© Галиаскаров Э.Г.,
Лабутина Т.В., 2010

© Ивановский государственный
химико-технологический
университет, 2010

ОГЛАВЛЕНИЕ

Введение	5
Лабораторная работа № 1. Пакет визуального моделирования Simulink.....	6
1. Общие сведения о пакете MATLAB	6
2. Начало работы	8
3. Выполнение заданий.....	12
4. Задачи для самостоятельного выполнения	22
4. Вопросы для самоконтроля знаний	25
Лабораторная работа № 2. Детерминированные модели с непрерывным временем. D-схемы.....	26
1. Постановка задачи	26
2. Построение модели.....	26
3. Выполнение задания.....	27
4. Вопросы для самоконтроля знаний.....	28
Лабораторная работа №3. Математическое планирование эксперимента на математической модели системы	30
1. Описание модели	30
2. Планирование эксперимента	31
2.1. План полного факторного эксперимента.....	33
2.2. Композиционный план 2-го порядка.....	37
3. Выполнение поставленной задачи	39
3.1. Реализация m-сценариев для выполнения задачи.....	39
4. Варианты заданий для выполнения лабораторной работы	42
5. Вопросы для самоконтроля знаний.....	43
Лабораторная работа №4. Математическое моделирование химических процессов. Модели РИВ и РИС	45
1. Системный анализ гомогенного химического реактора.....	45
1.1. Математическое описание подсистемы «химическая реакция»	45
1.2. Математическое описание подсистемы «движение потока».....	47
1.3. Математическое описание гомогенного реактора.	48
2. Реализация математического описания в Simulink	52
3. Варианты заданий	53
4. Вопросы для самоконтроля знаний.....	53
Лабораторная работа № 5. Дискретно-детерминированные и дискретно- стохастические модели. F и P-схемы	55
1. Дискретно-детерминированные модели или F-схемы.....	55
2. Дискретно-стохастические модели или P-схемы	57
3. Примеры решения задач.....	58
4. Задачи для самостоятельного выполнения	65

5. Вопросы для самоконтроля знаний.....	71
Лабораторная работа № 6. Моделирование случайных величин, событий и процессов.....	72
1. Метод статистических испытаний	72
2. Моделирование дискретных случайных величин	73
3. Моделирование непрерывных случайных величин	74
4. Сбор статистических данных для получения оценок характеристик случайных величин	76
5. Определение количества реализаций при моделировании случайных величин.....	77
6. Примеры выполнения задач.....	80
7. Задачи для самостоятельной работы	84
8. Вопросы для самоконтроля знаний.....	85
Лабораторная работа № 7. Непрерывно-стохастические модели. Q-схемы	86
1. Одноканальная модель (с отказами) M/M/1.....	86
2. Многоканальная модель (с отказами) M/M/m	88
3. Пример решения.....	89
4. Задание	94
5. Вопросы для самоконтроля знаний.....	95
Лабораторная работа №8. Распознавание образов с обучением. Линейный дискриминантный анализ в системе STATISTICA.....	96
1. Теоретическое введение	96
2. Задание № 1. Классический пример Фишера	99
3. Задание № 2. Медицинская задача.....	105
4. Вопросы для самоконтроля знаний.....	105
Лабораторная работа №9. Решение задач с использованием нейронных сетей в <i>MATLAB</i>	107
1. Общие сведения о нейронных сетях	107
1.1. Как работает биологическая нейронная сеть	108
1.2. Искусственная нейронная сеть	109
1.2. Области применения нейронных сетей.....	113
2. Практическая реализация методологии нейронных сетей	115
2.1. Аппроксимация функций с использованием НС	122
2.2. Формирование нейросетевых моделей в Simulink.....	122
2.3. Классификация с помощью ИНС	123
2.4. Прогнозирование значений процесса, заданного функцией времени .	124
3. Вопросы для самоконтроля знаний.....	125
4. Приложение	126
Список используемых источников	128

ВВЕДЕНИЕ

Лабораторный практикум предназначен для студентов специальности «Информационные системы и технологии», занимающихся по дисциплинам «Функциональные математические модели» и «Моделирование систем».

Целью лабораторного практикума является закрепление теоретических знаний и получение практических навыков в области моделирования сложных систем.

Основной упор сделан на визуальном построении моделей, использовании системного подхода при разработке моделей.

Практикум включает в себя 9 лабораторных работ, знакомящих студентов с правилами работы в пакете визуального моделирования Simulink, принципами разработки моделей по математическим схемам, планированием модельных экспериментов, обработкой результатов экспериментов, в том числе с использованием статистических методов и нейронных сетей.

Каждая лабораторная работа содержит краткое, но достаточное для осмысленной работы теоретическое введение, подробные примеры создания моделей и работы с ними, перечень заданий для самостоятельной работы по вариантам, список вопросов для самоконтроля знаний.

ЛАБОРАТОРНАЯ РАБОТА № 1.

ПАКЕТ ВИЗУАЛЬНОГО МОДЕЛИРОВАНИЯ SIMULINK

ЦЕЛЬ РАБОТЫ: освоение технологии решения задач моделирования систем с помощью *Simulink*.

1. ОБЩИЕ СВЕДЕНИЯ О ПАКЕТЕ MATLAB

История существования пакета **MATLAB**, название которого происходит от словосочетания *Matrix Laboratory* (матричная лаборатория) насчитывает уже более двух десятков лет. Таким образом, можно считать, что развитие **MATLAB** «шло в ногу» с развитием средств вычислительной техники: от «больших» ЭВМ с маленькими интерактивными возможностями до настольных компьютеров, позволяющих использовать в работе все пять (а иногда и шесть) способов восприятия информации. И, несмотря на достаточно высокую скорость смены поколений вычислительной техники, **MATLAB** успевал впитывать все наиболее ценное от каждого из них.

В результате к настоящему времени **MATLAB** представляет собой весьма удачное сочетание возможностей математики с последними достижениями в области вычислительной техники.

Одно из основных достоинств пакета состоит в том, что для работы пользователю достаточно знать о нем ровно столько, сколько требует решаемая задача. Так, в простейшем случае **MATLAB** может сыграть роль обыкновенного калькулятора, для использования которого достаточно помнить знаки математических операций. Если же решаемая задача требует создания каких-либо специальных инструментов, **MATLAB** предоставляет в распоряжение пользователя практически универсальный язык объектно-ориентированного программирования в сочетании с интерактивными средствами отладки создаваемых программ.

И все-таки, в первую очередь **MATLAB** — это средство математического моделирования, обеспечивающее проведение исследований практически во всех известных областях науки и техники. При этом структура пакета позволяет эффективно сочетать оба основных подхода к созданию модели: аналитический и имитационный.

Именно в сфере математического моделирования **MATLAB** позволяет наиболее полно использовать все современные достижения компьютерных технологий, в том числе средства визуализации и аудификации (озвучивания) данных, а также возможности обмена данными по сети Internet. Кроме того, пользователь имеет возможность создавать средствами **MATLAB** собственный графический интерфейс, отвечающий как его вкусам, так и требованиям решаемой задачи.

Как следует из названия пакета, он ориентирован в первую очередь на обработку массивов данных (матриц и векторов). Это позволило его разработчикам существенно повысить эффективность процедур, работающих с указанными типами данных, по сравнению с языками программирования «общего назначения» (*Pascal*, *C* и т. п.).

С точки зрения пользователя **MATLAB** представляет собой богатейшую библиотеку функций, единственная проблема работы с которой заключается в умении быстро отыскать те из них, которые нужны для решения данной задачи.

Для облегчения поиска библиотека функций разбита на разделы. Те из них, которые носят более общий характер и используются наиболее часто, входят в состав ядра **MATLAB**. Другие функции, которые являются специфическими для конкретной области, включены в состав соответствующих специализированных разделов. Эти разделы называются в **MATLAB** *Toolboxes* (инструменты). Каждый из них имеет свое собственное название, отражающее его предназначение. Полная комплектация пакета **MATLAB** содержит около 30 инструментальных приложений. В их число входят как достаточно стандартные для математических пакетов средства (решения дифференциальных и алгебраических уравнений, интегрального исчисления, символьных вычислений и т. д.), так и нетрадиционные, способные претендовать на определенную уникальность в своем роде: средства цифровой обработки изображений, поиска решений на основе нечеткой логики, аппарат построения и анализа нейронных сетей, средства финансового анализа и целый ряд других. Кроме того, имеются средства взаимодействия с популярными офисными продуктами фирмы **Microsoft** — **MS Word** и **MS Excel**.

Особое место среди инструментальных приложений занимает система визуального моделирования **Simulink**. В определенном смысле **Simulink** можно рассматривать как самостоятельный продукт фирмы Math Works (который даже в некоторых случаях продается в «именной» упаковке), однако он работает только при наличии ядра **MATLAB** и использует многие функции, входящие в его состав.

Необходимо отметить, что в **MATLAB** использована технология ассоциативной обработки файлов, поддерживаемая операционной системой **Windows**. Она заключается в том, что каждому типу файлов ставится в соответствие (ассоциируется с ним) определенное приложение, обеспечивающее обработку хранящихся в нем данных. Чтобы активизировать ассоциированное приложение, пользователю достаточно дважды щелкнуть на пиктограмме файла кнопкой мыши. Например, при выборе файла с расширением **.doc** загружается текстовый редактор **MS Word**.

В **MATLAB** используется несколько типов файлов, для каждого из которых определен свой допустимый набор операций и реализующие их средства. При работе с **Simulink** в основном используются файлы трех типов:

- М-файлы (имеющие расширение **.m**) — файлы, содержащие текст программы на языке **MATLAB**; в виде М-файлов реализованы все библиотечные функции **MATLAB**; по умолчанию М-файлы открываются с помощью собственного редактора/отладчика **MATLAB**;

- Мdl-файлы (файлы с расширением **.mdl**) — файлы моделей **Simulink**; могут быть открыты либо с помощью **Simulink** (в виде графического окна с блок-диаграммой), либо с помощью редактора/отладчика **MATLAB**;

- МАТ-файлы (с расширением **.mat**) — файлы, содержащие данные в двоичном коде; доступ к ним возможен либо из командного окна **MATLAB**, либо с помощью специальных средств **Simulink**.

Разработка моделей средствами **Simulink** (в дальнейшем S-моделей) основана на использовании технологии *drag-and-drop* («перетаски и оставь»). В ка-

честве «кирпичиков» для построения S-модели используются модули (или блоки), хранящиеся в библиотеке *Simulink*.

Simulink хорош тем, что, с одной стороны, обеспечивает пользователю доступ ко всем основным возможностям пакета *MATLAB*, а с другой — является достаточно самостоятельной его компонентой, в том смысле, что при работе с ним не обязательно иметь навыки в использовании других инструментов, входящих в состав пакета.

Блоки, включаемые в создаваемую модель, могут быть связаны друг с другом как по информации, так и по управлению. Тип связи зависит от типа блока и логики работы модели. Данные, которыми обмениваются блоки, могут быть скалярными величинами, векторами или матрицами произвольной размерности.

Любая S-модель может иметь иерархическую структуру, то есть состоять из моделей более низкого уровня, причем число уровней иерархии практически не ограничено.

Наряду с другими параметрами моделирования пользователь может задавать способ изменения модельного времени (с постоянным или переменным шагом), а также условия окончания моделирования.

В ходе моделирования имеется возможность наблюдать за процессами, происходящими в системе. Для этого используются специальные «смотровые окна», входящие в состав библиотеки *Simulink*. Интересующие пользователя характеристики системы могут быть представлены как в числовой, так и в графической форме. Кроме того, существует возможность включения в состав модели средств анимации.

Еще одно важное достоинство *Simulink* заключается в том, что он является открытой системой: состав библиотеки может быть пополнен пользователем за счет разработки собственных блоков.

Все примеры и задания, представленные в практикуме, реализованы на базе *MATLAB* версии 7.20.232 (R2006a).

2. НАЧАЛО РАБОТЫ

Запуск *Simulink* можно произвести одним из трех способов:

- нажав соответствующую кнопку на панели инструментов командного окна *MATLAB*;
- введя команду *Simulink* в активной строке командного окна;
- выбрав команду *New\Model* в меню *File*.

Использование первого и второго способов приводит к открытию окна браузера разделов библиотеки *Simulink* (рис.1.1), а по команде *New\Model* кроме него открывается еще и пустое окно для создания S-модели (рис.1.2).

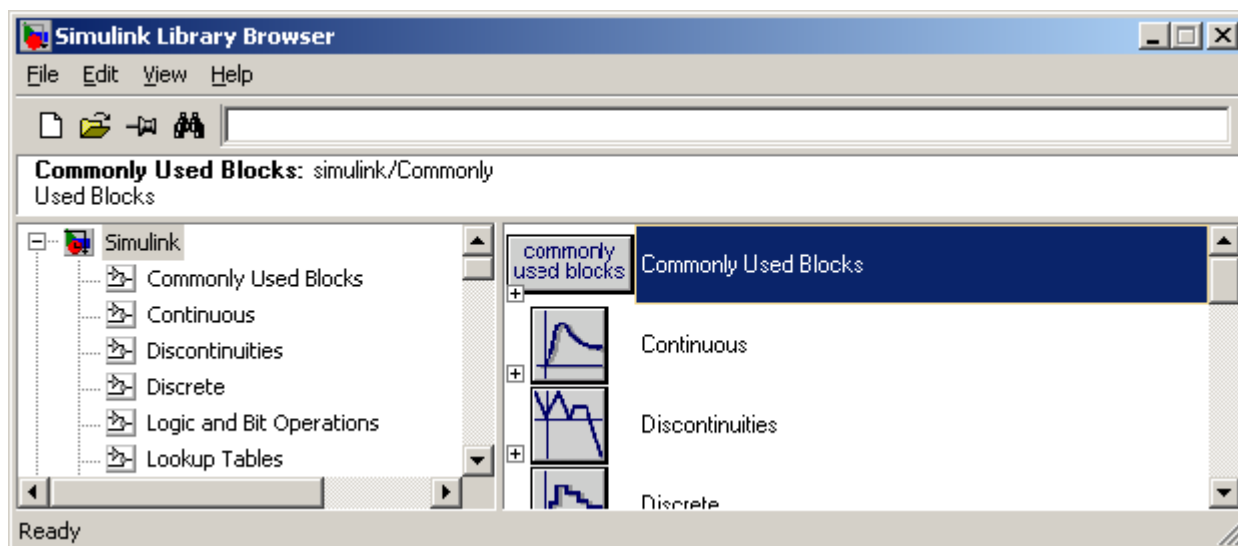


Рис.1.1. Окно браузера разделов библиотеки *Simulink*

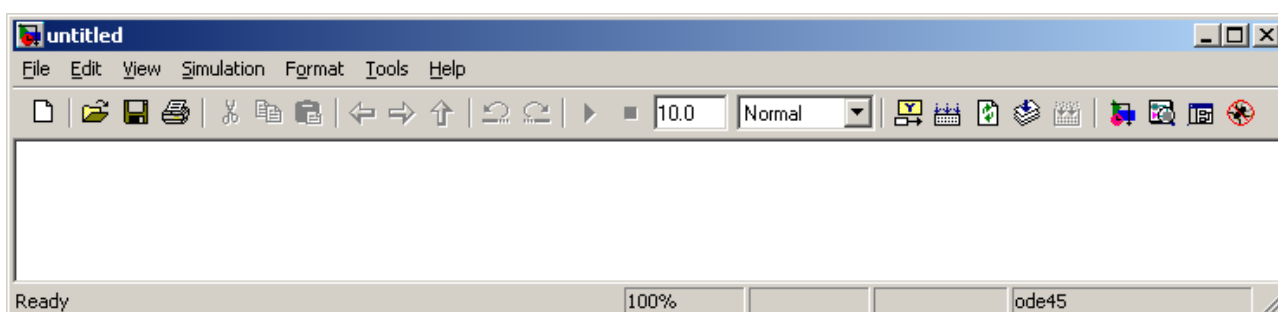


Рис.1.2. Пустое окно для создания блок-диаграммы модели

Окно браузера разделов содержит панель инструментов, собственно список разделов, реализованный в виде дерева, и два вспомогательных подокна, одно из которых используется для вывода комментария к выбранному в списке элементу, а другое — для представления пиктограммы этого элемента (рис.1.1).

Кнопки панели инструментов браузера имеют следующее предназначение (рис.1.3):



Рис.1.3. Кнопки панели инструментов окна браузера *Simulink*

- 1 — создать новую S-модель (точнее, открыть новое окно модели);
- 2 — открыть одну из существующих S-моделей;
- 3 — изменить свойство окна браузера «Всегда сверху»; данная кнопка работает как переключатель: при ее «нажатии» связанное с ней значение изменится на противоположное, при этом изменяется и изображение кнопки; если данное свойство установлено, то окно браузера отображается всегда поверх других открытых окон (в исходном состоянии окно браузера этим свойством не обладает);

4 — найти блок, название которого (или первые несколько символов названия) указано в расположенном справа от кнопки текстовом поле; если требуемый блок найден, в окне браузера открывается соответствующий раздел библиотеки, а блок отображается как выбранный, в противном случае в подокне комментария выводится сообщение *Not found* <имя блока> (Не найден).

Список разделов библиотеки *Simulink* представлен в основном окне браузера в виде дерева, и правила взаимодействия с ним являются стандартными для списков такого типа:

- пиктограмма свернутого узла дерева содержит символ "+", а пиктограмма развернутого — символ "-";
- чтобы развернуть или свернуть, узел дерева, достаточно щелкнуть на его пиктограмме ЛКМ (или дважды щелкнуть на текстовой метке узла).

Библиотека *Simulink* имеет следующую структуру (рис.1.4).

На верхнем уровне иерархии расположена основная библиотека, которая носит соответствующее название — *Simulink*, а также внушительный список других библиотек, содержащих наборы блоков, ориентированных на решение специализированных задач в разных областях моделирования.

Ко второму уровню относятся разделы основной библиотеки и дополнительных библиотек. И, наконец, нижний уровень иерархии образуют собственно блоки *Simulink*, которые и играют роль «кирпичиков» при построении S-модели. Чтобы вставить блок в S-модель, необходимо выбрать его в списке (щелкнув ЛКМ на графической или текстовой метке блока) и, не отпуская кнопку мыши, перетащить его контур в поле блок-диаграммы. Что из этого должно получиться, вы узнаете немного позже. А пока вернемся к упомянутому окну блок-диаграммы, общий вид которого был показан на рис.1.2.

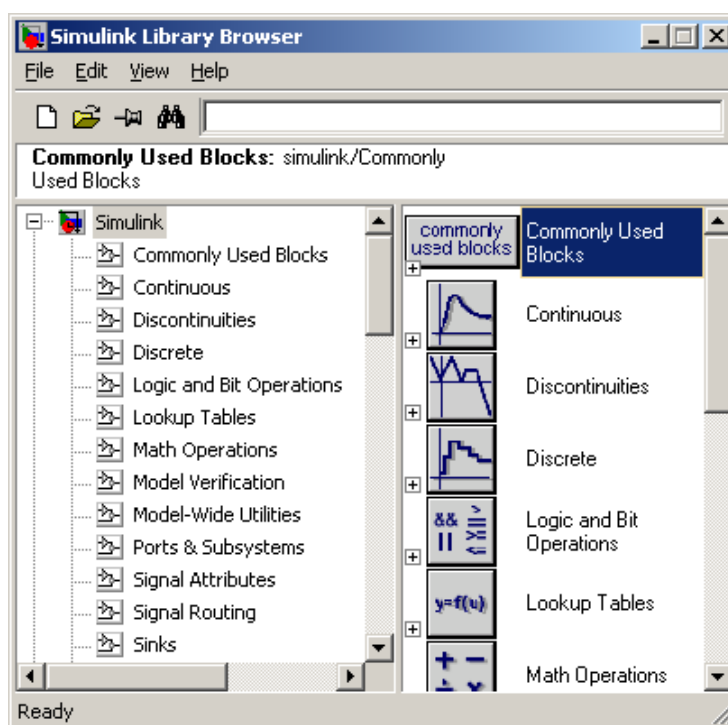


Рис.1.4. Окно браузера разделов библиотеки *Simulink*

С точки зрения пользовательского интерфейса оно представляет собой обычное окно со всеми вытекающими из этого последствиями (его можно сворачивать, изменять размеры окна, перемещать по экрану и т.д.). В отличие от окна браузера библиотеки, закрытие окна блок-диаграммы не приводит к выходу из *Simulink*. Меню окна блок-диаграммы содержит следующие разделы:

- *File* — команды работы с mdl-файлами;
- *Edit* — команды редактирования блок-диаграммы и опции для работы с библиотекой;
- *View* — команды изменения формата окна и управления выводом дополнительной информации;
- *Simulation* — команды управления моделированием,
- *Format* — команды редактирования формата (т. е. внешнего облика) блоков диаграммы и блок-диаграммы в целом,
- *Tools* – набор инструментов, используемых при организации и выполнении экспериментов с моделями

На панель инструментов выведены следующие команды меню (рис.1.5):

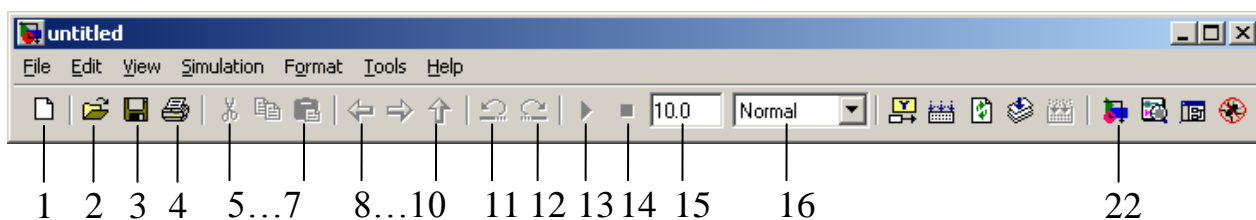


Рис.1.5. Панель инструментов окна блок-диаграммы

Перечислим наиболее важные из них:

- 1 — *New Model* — открыть новое (пустое) окно блок-диаграммы;
- 2 — *Open Model* — открыть существующий mdl-файл;
- 3 — *Save Model* — сохранить mdl-файл на диске;
- 4 — *Print Model* — вывод на печать блок-диаграммы;
- 5...7 — команды редактирования блок-диаграммы (вырезать, копировать, вставить);
- 8...10 — навигация по модели;
- 11 — *Undo* — отменить предыдущую операцию редактирования;
- 12 — *Redo* — восстановить результат отмененной операции редактирования;
- 13 — *Start Simulation* — кнопка запуска модели на исполнение;
- 14 — *Stop simulation* — кнопка остановки исполнения модели;
- 15 — *Simulation Stop Time* — задает время завершения исполнения модели;
- 16 — режим выполнения модели;
- 22 — *Library Browser* — активизировать окно браузера библиотеки.

Краткие комментарии к кнопкам панели инструментов, а также к пунктам меню окна блок-диаграммы отображаются в левой части строки состояния, когда указатель мыши находится над соответствующим элементом интерфейса. Это же текстовое поле используется и для индикации состояния *Simulink*: *Ready* (Готов) или *Running* (Выполнение). Наряду с указанной информацией в строке состояния отображаются:

- установленный масштаб просмотра блок-диаграммы (в процентах, исходное значение равно 100%);
- индикатор степени завершенности сеанса моделирования («включается» только при запуске модели);
- текущее значение модельного времени (выводится также только после запуска модели);
- используемый алгоритм расчета состояний модели.

Применение команд меню, не выведенных на панель инструментов, будет рассмотрено немного позже. Сейчас остановимся на описании основных возможностей *Simulink*.

Основная библиотека блоков разбита на 16 разделов, содержимое которых не может изменяться пользователем (количество разделов, их наименование и состав может меняться от версии к версии):

- **Commonly Used Blocks** (Наиболее часто используемые блоки. Раздел динамически изменяется);
- **Continuous** (Блоки для моделирования непрерывных систем);
- **Discontinuities** (Блоки для дискретизации непрерывных сигналов);
- **Discrete** (Блоки для моделирования дискретных систем);
- **Logic and Bit Operations** (Блоки логических и битовых операций);
- **Math Operations** (Математические блоки);
- **Ports and Subsystems** (Блоки для создания подсистем);
- **Signal Routing** (Блоки для управления потоками сигналов);
- **Sinks** (Блоки-«получатели»);
- **Sources** (Блоки-«источники»);
- **User-Defined Functions** (Функции, определяемые пользователем);

Функции и работу блоков, входящих в эти разделы, будем рассматривать по мере выполнения конкретных заданий.

3. ВЫПОЛНЕНИЕ ЗАДАНИЙ

Перед выполнением заданий необходимо загрузить пакет *MATLAB* и, выбрав команду *New\Model* в меню *File*, запустить окно браузера разделов библиотеки *Simulink* и пустое окно для создания S-модели. Используя технологию *drag-and-drop*, переместить необходимые блоки из разделов библиотеки в пустое окно S-модели.

Для выполнения заданий нам потребуются следующие блоки из раздела **Sources**:

- **Step** – генерирует единичный дискретный сигнал с заданными параметрами;


- **Constant** – формирует постоянную величину (скаляр, вектор или матрицу);
- **Clock** – источник непрерывного временного сигнала;
- **Pulse Generator** – генератор импульсных сигналов;
- **Random Number** – источник дискретного сигнала, амплитуда которого является случайной величиной, распределенной по нормальному закону;
- **Uniform Random Number** – источник дискретного сигнала, амплитуда которого является равномерно распределенной случайной величиной,

и Sinks:

- **Display** – предназначен для отображения численных значений величин;
- **Scope** (осциллограф) – предназначен для отображения динамики изменения характеристик систем.

Задание 1. Формирование ступенек разной высоты и сдвига

Технология выполнения задания:

1. «Перетащите» с помощью мыши блок **Step** из раздела **Sources** и блок **Scope** из раздела **Sinks** в окно S-модели.
2. Соедините, используя мышь, выход блока **Step** с входом блока **Scope**;
3. Формирования единичной ступеньки:
 - двойным щелчком по блоку ЛКМ откройте окно *Block Parameters: Step*. Блок имеет 4 параметра настройки:
 - *Step time* (шаг времени) – определяет длительность сигнала;
 - *Initial value* (Начальное значение) – задает значение амплитуды сигнала в начальный момент времени;
 - *Final value* (Конечное значение) - задает значение амплитуды сигнала по истечении шага моделирования;
 - *Sample time* (Образец времени) -;
 - задайте параметры блока **Step**: *Step time* – 1; *Initial value* – 0; *Final value* – 1; *Sample time* – 0;
 - закройте окно *Block Parameters: Step*;
 - двойным щелчком по блоку **Scope** ЛКМ откройте окно *Scope*;
 - запустите выполнение модели, нажав на кнопку  Панели инструментов окна блок-диаграммы (рис.1.5).
4. Формирование единичной ступеньки в момент времени 5 с.:
 - повторите действия пункта 3, задав параметры блока **Step**: *Step time* – 5; *Initial value* – 0; *Final value* – 1; *Sample time* – 0.
5. Формирование единичного импульса. Задание единичного импульса подразумевает, что площадь импульса равна 1:
 - повторите действия пункта 3, задав параметры блока **Step**: *Step time* – 1; *Initial value* – 1; *Final value* – 0; *Sample time* – 0.

Задание 2. Формирование постоянных значений: скаляр, вектор, матрица

Технология выполнения задания.

Скаляр задается числом (5), вектор задается последовательностью чисел в квадратных скобках через пробел ([1 2 4 6]), матрица задается последовательностью векторов, разделенных точкой с запятой ([1 2 3; 5 6 7; 8 9 10]).

- 1.«Перетащите» с помощью мыши блок **Constant** из раздела **Sources** и блок **Display** из раздела **Sinks** в окно S-модели.
- 2.Соедините, используя мышью, выход блока **Constant** с входом блока **Display**.
- 3.Формирования скаляра:
 - двойным щелчком по блоку **Constant** ЛКМ откройте окно *Block Parameters: Constant*;
 - задайте любое числовое значение, например 5;
 - закройте окно *Block Parameters: Constant*;
 - запустите выполнение модели, нажав на кнопку 11 Панели инструментов окна блок-диаграммы (рис.1.5).
- 4.Формирования вектора:
 - задайте вектор, например [5 6 7 8 9];
 - после запуска выполнения модели на блоке **Display** в правом нижнем углу появится стрелка ▼, указывающая на необходимость увеличения окна. Щелкните ЛКМ по блоку и увеличьте размеры окна, потянув нажатой ЛКМ за один из углов блока (■).
- 5.Формирования матрицы:
 - задайте матрицу 3×3, например [5 6 7; 8 9 3; 3 4 5];
 - после запуска выполнения модели на блоке **Display** в правом нижнем углу появится стрелка ►, указывающая на необходимость увеличения окна. Щелкните ЛКМ по блоку и увеличьте размеры окна, потянув нажатой ЛКМ за один из углов блока (■).

Задание 3. Работа с источниками временного сигнала

Технология выполнения задания.

- 1.Для того чтобы задать время моделирования, выполните следующие действия:
 - в меню окна блок-диаграммы выберите вкладку *Simulation* и выполните команду *Simulation parameters* либо нажмите сочетание клавиш Ctrl+E;
 - в появившемся окне *Simulation parameters* выберете вкладку *Solver* и выставите значения в группе *Simulation time: Start time 0.0* и *Stop time 15.0*.
2. Для того чтобы контролировать текущее время моделирования, выполните следующие действия:

- «перетащите» с помощью мыши блоки **Clock**, **Pulse Generator**, **Random Number**, **Uniform Random Number** из раздела **Sources** и блок **Scope** из раздела **Sinks** в окно S-модели;
- соедините, используя мышь, выход блока **Clock** с входом блока **Scope**;
- запустите выполнение модели, сделайте выводы;
- повторите те же действия с остальными источниками сигнала.

Справка: Блок **Pulse Generator** имеет следующие параметры: тип импульса (*Pulse type*), амплитуда сигнала (*Amplitude*), величина периода сигнала (*Period*), ширина импульса (*Pulse width*), величина задержки сигнала (*Phase delay*), шаг изменения модельного времени (*Sample time*). Значения параметров 2...4 должны задаваться как целое число шагов модельного времени (*number of samples*). Поэтому выбор значений параметров блока целесообразно начинать с выбора величины *Sample time*. Размер шага можно указать как в форме константы, так и в форме вычисляемого выражения. Если вычисленное значение является дробным, оно округляется до целого. Аналогично может быть задано и значение амплитуды, но в случае дробной величины округление не производится.


Блок **Random Number** обеспечивает формирование сигналов, амплитуда которых является случайной величиной, распределенной по нормальному закону с заданными параметрами. Блок имеет четыре параметра настройки. Первые два — *Mean* и *Variance* — являются параметрами нормального закона (среднее и дисперсия), третий — *Initial seed* — задает начальное значение для инициализации генератора последовательности случайных чисел. При фиксированном значении этого параметра генератор всегда вырабатывает одну и ту же последовательность СЧ. Четвертый параметр (*Sample time*) используется так же, как и в рассмотренных ранее блоках.

Блок **Uniform Random Number** обеспечивает формирование сигналов, амплитуда которых является случайной величиной, распределенной равномерно в заданном интервале. В блоке имеются такие же параметры настройки *Initial seed* и *Simple time*, как и в *Random Number*. Еще два параметра (*Minimum* и *Maximum*) задают диапазон распределения СВ.

Задание 4. Работа с блоком Scope. Представление данных в графическом виде

Технология выполнения задания.


1. Вывод в различных экранах блока **Scope** несколько графиков:

- «перетащите» блоки **Pulse Generator** и **Sine Wave** из раздела **Sources** и блок **Scope** из раздела **Sinks** в окно S-модели;
- двойным щелчком ЛКМ откройте окно блока **Scope**, нажмите иконку *Properties* , в появившемся окне установите количество отобра-

ражаемых графиков *Number of axes* – 2, нажмите *Apply* и закройте окно;

- на блоке **Scope** появятся два входа, которые необходимо соединить с соответствующими источниками сигналов;
- запустите выполнение модели, сделайте выводы.

2. Вывод нескольких графиков на одном экране блока **Scope**:

- двойным щелчком ЛКМ откройте окно блока **Scope**, нажмите иконку *Properties* , в появившемся окне установите количество отображаемых графиков *Number of axes* – 1, нажмите *Apply* и закройте окно;
- «перетащите» блок **Mux** из раздела **Signal Routing** (назначение и свойства блока изучите самостоятельно);
- двойным щелчком ЛКМ откройте окно блока **Mux** и установите число входов *Number of input* равным 2, нажмите *Apply* и закройте окно;
- соедините выходы блоков-источников сигналов с входами блока **Mux**, а выход блока **Mux** с входом блока **Scope**;
- запустите модель, сделайте выводы.

Задание 5. Решение простейших математических задач (раздел **Math**)

Технология выполнения задания.

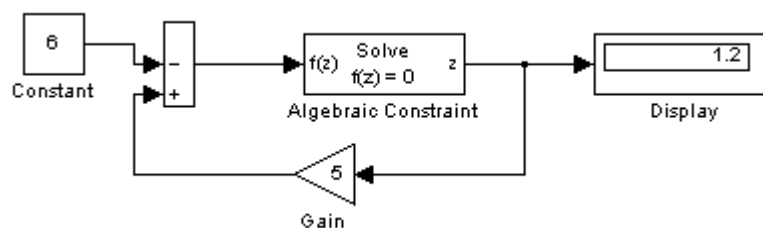
1. Работа с блоком **Sign**: проверяет знак входного сигнала и формирует одно из 3-х значений: -1 – если входной сигнал < 0 ; 0 - если входной сигнал = 0; 1 – если входной сигнал > 1 :
 - «перетащите» блок **Sign** из раздела **Math Operations**, блок **Sine wave** из раздела **Source**, блок **Scope** из раздела **Sinks**;
 - соедините блоки **Sine wave** – **Sign** – **Scope**;
 - откройте окно блока **Scope** и запустите модель, сделайте выводы.
2. Выполнение арифметических действий. Найдите значение данного выражения $3+2-2+5$:
 - «перетащите» 4 блока **Constant** из раздела **Sources**, блок **Sum** из раздела **Math Operations** и блок **Display** из раздела **Sinks**;
 - двойным щелчком ЛКМ откройте окно параметров каждого из блоков **Constant** и установите соответствующие значения констант;
 - двойным щелчком ЛКМ откройте окно параметров блока **Sum**, измените свойство *Icon shape* (форма иконки) на *rectangular* (прямоугольник), введите в поле *List of signs* (список знаков) последовательность $++-+$, отражающую последовательность арифметических операций в выражении, нажмите кнопку *Apply* и закройте окно параметров;
 - измените размеры блока **Sum**, если необходимо;
 - соедините выход блока **Constant** с входом блока **Sum**, отмеченным соответствующим знаком;
 - соедините выход блока **Sum** с входом блока **Display**;

- запустите модель, на экране блока **Display** появится результат выражения $= 8$.
3. Выполнение операции умножения и деления. Для выполнения этого задания Вам понадобятся блоки **Product** и **Gain** из раздела **Math Operations**. С работой данных блоков ознакомьтесь самостоятельно.

Справка: Блок **Product** имеет два параметра: число входов (*Number of inputs*) и умножение (*Multiplication*). Параметр *Number of inputs* может иметь числовое значение (только для операции умножения), либо последовательность знаков умножения и деления (например * * / *). Параметр *Multiplication* может принимать только два значения: Element-wise (операции со скалярными значениями) или Matrix (операции с матрицами).

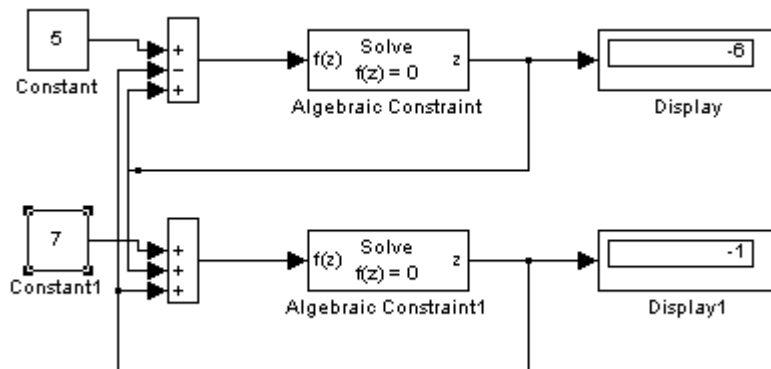
Блок **Gain** имеет два параметра: множитель (*Gain*) и умножение (*Multiplication*). Параметр *Gain* задает постоянный множитель, на который будет умножаться входной сигнал. Параметр *Multiplication* может принимать только три значения: Element-wise ($K \cdot u$) (умножение на скалярную величину), Matrix ($K \cdot u$) (умножение постоянного множителя на матрицу) и Matrix ($K \cdot u$) (умножение матрицы на постоянный множитель).

4. Отыскание корней алгебраических уравнений. Пусть необходимо решить уравнение вида $5x - 6 = 0$:
- «перетащите» в окно S-модели следующие блоки: **Constant**, **Sum**, **Display**, **Gain** и **Algebraic Constraint** из раздела **Math Operations**;
 - установите на блоке **Constant** значение: 6; на блоке **Gain** значение: 5; на блоке **Sum** параметры: rectangular и + -;
 - соедините выход блока **Constant** с входом **Sum**, отмеченным знаком «-»;
 - соедините выход блока **Sum** с входом **Algebraic Constraint**;
 - соедините выход блока **Algebraic Constraint** с входом блока **Sum**, отмеченным знаком «+»;
 - поместите блок **Gain** на линию соединения выхода блока **Algebraic Constraint** с входом блока **Sum** (блок автоматически встроится в линию связи);
 - соедините выход блока **Algebraic Constraint** с входом блока **Display**;
 - запустите модель.

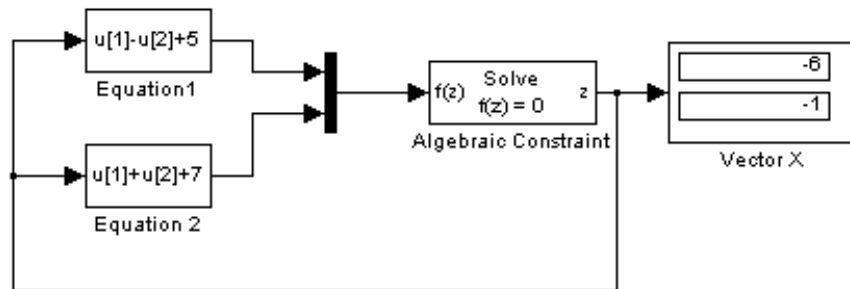


5. Решение систем алгебраических уравнений производится аналогично, необходимо только сформировать столько конструктивных линий, как это сделано в пункте 4, сколько уравнений содержится в системе. Решите, например, систему, представленную ниже:

$$\begin{cases} x - y + 5 = 0 \\ x + y + 7 = 0 \end{cases}$$

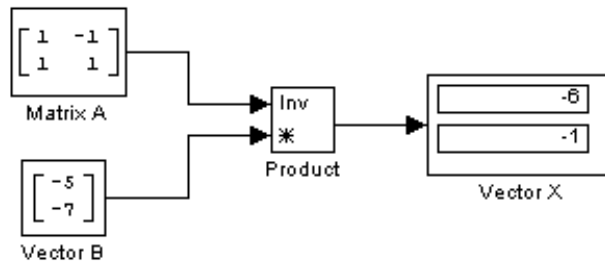


Решить систему линейных уравнений можно еще проще, учитывая, что количество уравнений может быть большим.



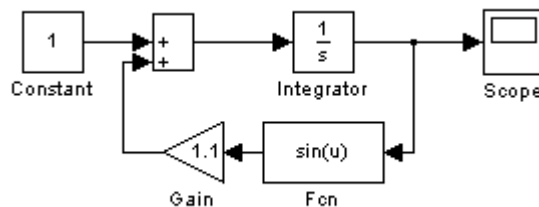
Справка: Блок **Fcn** из раздела **User-Defined Functions** имеет единственный параметр: **Expression** (выражение). В качестве переменной используйте переменную **u**, если таких переменных несколько, то они различаются индексами: **u[1]**, **u[2]**, и т.п. Можно записать любое математическое выражение в виде, допустимом для процедурных языков программирования, например: $\sin(u)$; $\exp(-u[1])*u[2]^2-3.5*u[3]$

Поскольку **MATLAB** отлично справляется с векторной алгеброй, самым простым и очевидным решением будет матричный способ, реализующий аналитическое решение:

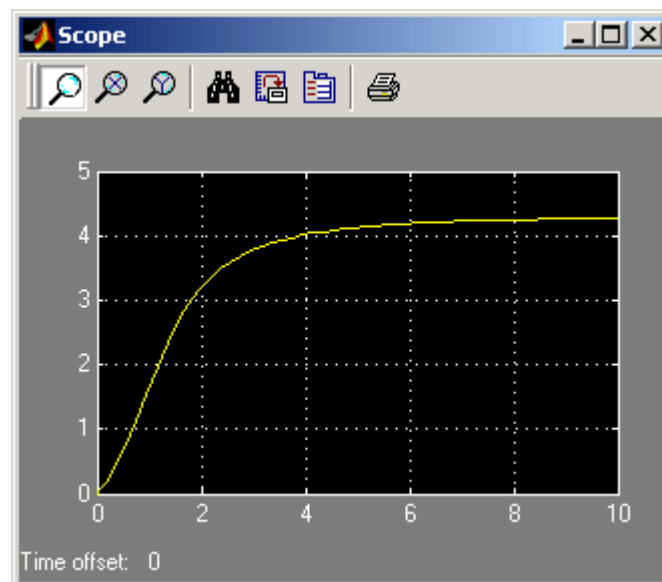


6. Решение обыкновенных дифференциальных уравнений (о.д.у.). В качестве примера решим уравнение вида: $\frac{dy}{dt} = 1.1\sin(y) + 1$ для начального условия $y(t=0) = 0$.

Попробуйте сделать это задание самостоятельно, руководствуясь рисунком.

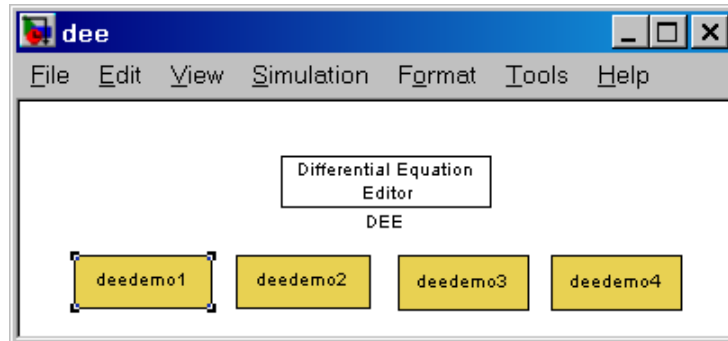


Результатом решения будет график функции, изображенный на рисунке.

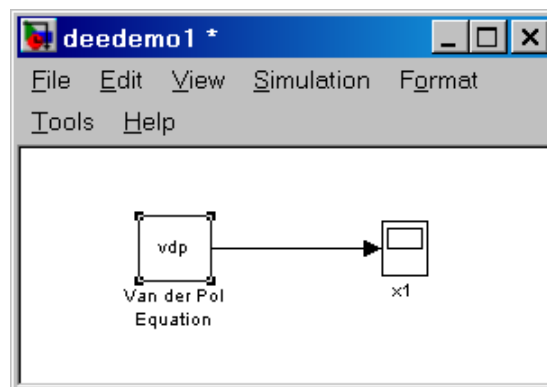


Справка: В блоке **Integrator** из раздела **Continuous** нас интересует пока только один параметр **Initial condition** (Начальное условие). Если таких условий несколько, они записываются через пробел в квадратных скобках, например: [1.5 0 0].

Для решения обыкновенных дифференциальных уравнений и их систем в *MATLAB* имеется специальный редактор дифференциальных уравнений. Редактор отсутствует среди библиотеки блоков *Simulink*. Для вызова редактора необходимо в командной строке *MATLAB* набрать команду: **dee**. В результате появится окно, содержащее четыре примера.



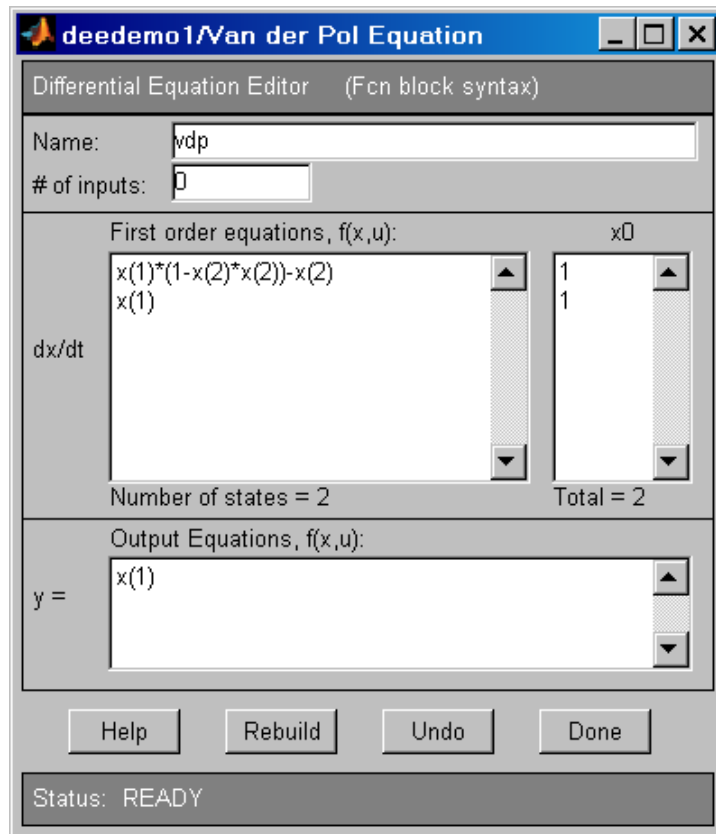
Необходимо выбрать любой из примеров и открыть его. Откроется обычная S-модель, как показано на рисунке ниже.



Перетащите левый блок (это и есть редактор) на свою модель и откройте редактор для изменения параметров в соответствии с Вашими потребностями.

Поля редактора имеют следующее значение:

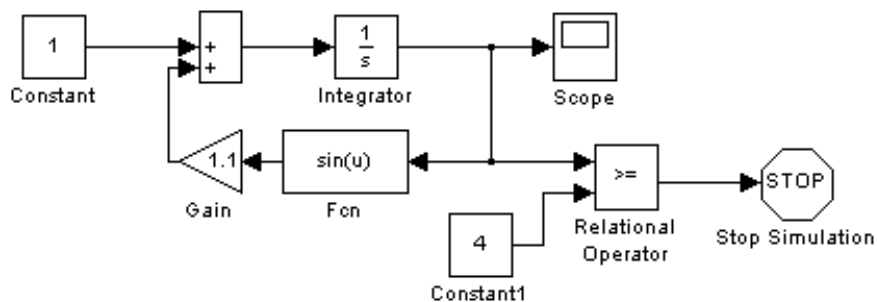
- **Name** – наименования блока;
- **# of inputs** – количество входов (по умолчанию 0);
- **dx/dt** – собственно дифференциальные первого порядка;
- **x0** – начальные значения переменных уравнения;
- **y** – выходные значения (результат)



Задание 6. Управление логикой и временем выполнения модели

Технология выполнения задания.

1. За основу возьмем модель из задания 5, пункта 6. Потребуем, чтобы вычисление прекратилось в момент, когда значение искомой функции достигнет 4. Для этого изменим нашу модель следующим образом:
 - добавьте к модели блок **STOP** из раздела **Sinks**, блок **Rational Operator** из раздела **Logic and Bit Operations**, еще один блок **Constant** из раздела **Sources**;
 - соедините блоки как показано на схеме;
 - установите конечное время моделирования в **inf**;
 - откройте окно блока **Scope** и запустите модель, сделайте выводы.



4. ЗАДАЧИ ДЛЯ САМОСТОЯТЕЛЬНОГО ВЫПОЛНЕНИЯ

1. Решить систему линейных уравнений:

- аналитическим методом $AX = B$ (где A и B – соответственно матрица коэффициентов при неизвестных и вектор свободных членов системы);
- методом последовательных приближений $AX - B = 0$ (с использованием блоков **Fcn** и с использованием блоков **Constant**, **Sum**, **Product** или **Gain** и одного блока **Algebraic Constraint**).

Номер варианта соответствует номеру компьютера.

Таблица 1.1

Вариант	Система линейных уравнений
1	$\begin{cases} 2,7x_1 + 14,9x_2 + 1,3x_3 = 2,1 \\ 1,3x_1 + 2,8x_2 - 9,3x_3 = 1,7 \\ 15,8x_1 + 4,1x_2 - 1,7x_3 = 0,8 \end{cases}$
2	$\begin{cases} 0,34x_1 - 7,9x_2 + 0,3x_3 = -4,1 \\ 11,3x_1 + 2,8x_2 - 1,3x_3 = 0,7 \\ 1,8x_1 + 2,4x_2 + 7,7x_3 = 0,48 \end{cases}$
3	$\begin{cases} -1,56x_1 + 1,9x_2 + 12,3x_3 = 4,1 \\ 6,3x_1 + 0,8x_2 - 0,3x_3 = -21,2 \\ 0,28x_1 + 8,1x_2 - 1,2x_3 = 3,8 \end{cases}$
4	$\begin{cases} 2,66x_1 + 0,9x_2 + 15,3x_3 = 7,1 \\ 1,3x_1 + 22,8x_2 + 3,3x_3 = 6,89 \\ 15,8x_1 - 4,1x_2 - 1,7x_3 = 2,8 \end{cases}$
5	$\begin{cases} 0,454x_1 + 9,67x_2 + 2,3x_3 = 2,167 \\ 1,3x_1 + 0,8x_2 - 9,4x_3 = 3,789 \\ 13,8x_1 + 2,1x_2 - 1,7x_3 = 1,845 \end{cases}$
6	$\begin{cases} 2,56x_1 - 4,9x_2 - 18,3x_3 = 21,1 \\ 21,3x_1 - 2,8x_2 - 5,3x_3 = 31,8 \\ 4,8x_1 + 41,1x_2 - 2,24x_3 = 9,8 \end{cases}$
7	$\begin{cases} 2,23x_1 + 14,8x_2 + 1,3x_3 = 7,1 \\ 1,35x_1 - 2,82x_2 - 19,3x_3 = 7 \\ 17,5x_1 + 4,1x_2 - 1,7x_3 = 6,8 \end{cases}$
8	$\begin{cases} 5,4x_1 - 11,8x_2 + 1,3x_3 = -12,1 \\ 1,3x_1 + 0,8x_2 - 9,3x_3 = 13,8 \\ 11,3x_1 - 4,1x_2 - 1,7x_3 = 10,85 \end{cases}$

Вариант	Система линейных уравнений
9	$\begin{cases} 0,34x_1 - 0,9x_2 + 9,3x_3 = -13,1 \\ 13,2x_1 + 2,8x_2 - 1,3x_3 = 0,999 \\ 1,2x_1 + 7,4x_2 + 0,7x_3 = 0,482 \end{cases}$
10	$\begin{cases} 3,6x_1 + 1,9x_2 - 13,3x_3 = 6,1 \\ 1,3x_1 - 12,8x_2 + 2,3x_3 = 4,89 \\ 10,8x_1 - 4,1x_2 + 1,7x_3 = 2,34 \end{cases}$
11	$\begin{cases} 0,49x_1 + 0,67x_2 + 12,3x_3 = -7,1 \\ 0,3x_1 + 10,5x_2 - 1,4x_3 = 3,12 \\ 10,8x_1 + 1,89x_2 - 1,3x_3 = 1,39 \end{cases}$
12	$\begin{cases} 2,06x_1 + 14,9x_2 - 1,3x_3 = 2,11 \\ 17,3x_1 - 6,8x_2 + 2,3x_3 = 3,5 \\ 0,8x_1 - 4,1x_2 - 22,4x_3 = 7,8 \end{cases}$
13	$\begin{cases} 5,6x_1 - 0,9x_2 - 15,3x_3 = 1,1 \\ 12,3x_1 - 2,8x_2 - 5,22x_3 = 3,1 \\ -3,8x_1 + 21,2x_2 - 1,24x_3 = 5,8 \end{cases}$
14	$\begin{cases} 10,56x_1 - 1,4x_2 + 1,13x_3 = 2,167 \\ -1,3x_1 + 0,8x_2 + 9,9x_3 = -3,89 \\ 0,128x_1 - 3,1x_2 - 0,2x_3 = 0,8 \end{cases}$
15	$\begin{cases} 21,6x_1 - 4,9x_2 + 1,3x_3 = 25,1 \\ 1,3x_1 + 2,5x_2 + 19,4x_3 = 1,55 \\ 5,8x_1 + 24,1x_2 - 1,9x_3 = 2,56 \end{cases}$
16	$\begin{cases} 2,22x_1 - 12,6x_2 + 4,2x_3 = 5,1 \\ 9,3x_1 - 2,8x_2 - x_3 = 3,11 \\ 5,8x_1 + 4,12x_2 - 9,7x_3 = -0,8 \end{cases}$

2. Решить систему о.д.у. для начальных условий: $y_1(0) = 2$; $y_2(0) = 4$; $y_3(0) = 0$.

Таблица 1.2

№	Система о.д.у.	№	Система о.д.у.
1	$\begin{cases} \frac{dy_1}{dt} = -0.8y_1y_2 + 0.4y_3 \\ \frac{dy_2}{dt} = -0.8y_1y_2^2 + 0.8y_3 \\ \frac{dy_3}{dt} = 0.6y_1y_2 - 0.4y_3 \end{cases}$	9	$\begin{cases} \frac{dy_1}{dt} = 2y_1 - y_2 + y_3 \\ \frac{dy_2}{dt} = y_1 + 2y_2 - y_3 \\ \frac{dy_3}{dt} = y_1 - y_2 + 2y_3 \end{cases}$

№	Система о.д.у.	№	Система о.д.у.
2	$\frac{dy_1}{dt} = 3y_1 - y_2 + y_3$ $\frac{dy_2}{dt} = y_1 + y_2 + y_3$ $\frac{dy_3}{dt} = 4y_1 - y_2 + 4y_3$	10	$\frac{dy_1}{dt} = 2y_1 - y_2 - y_3$ $\frac{dy_2}{dt} = y_1 + 2y_3$ $\frac{dy_3}{dt} = y_1 - 2y_2 - y_3$
3	$\frac{dy_1}{dt} = y_1 - 2y_2 - 2y_3$ $\frac{dy_2}{dt} = y_1 - 2y_2 + 2y_3$ $\frac{dy_3}{dt} = 3y_1 - 3y_2 + 5y_3$	11	$\frac{dy_1}{dt} = 3y_1 - 2y_2 - y_3$ $\frac{dy_2}{dt} = 3y_1 - 4y_2 - 3y_3$ $\frac{dy_3}{dt} = 2y_1 - 4y_2$
4	$\frac{dy_1}{dt} = 2y_1 + y_2^2 - 1$ $\frac{dy_2}{dt} = 6y_1 - y_2^2 + 1$	12	$\frac{dy_1}{dt} = -4y_1^2 + y_2^2$ $\frac{dy_2}{dt} = 4y_2 - 8$
5	$\frac{dy_1}{dt} = 4 - 4y_1 - 2y_2$ $\frac{dy_2}{dt} = y_1 y_2$	13	$\frac{dy_1}{dt} = 2 - y_1^2 + y_2$ $\frac{dy_2}{dt} = 2y_1(y_1 - y_2)$
6	$\frac{dy_1}{dt} = y_1^2 + y_2^2 - 6y_1 - 8y_2$ $\frac{dy_2}{dt} = 9 - (y_1 - 2y_2)^2$	14	$\frac{dy_1}{dt} = \ln(2 - y_2^2)$ $\frac{dy_2}{dt} = e^{y_1} - e^{y_2}$
7	$\frac{dy_1}{dt} = \ln\left(\frac{y_2^2 - y_2 + 1}{3}\right)$ $\frac{dy_2}{dt} = y_1^2 - y_2^2$	15	$\frac{dy_1}{dt} = 2y_1 - y_2 - y_3$ $\frac{dy_2}{dt} = 2y_1 - y_2 - 2y_3$ $\frac{dy_3}{dt} = -y_1 + y_2 + 2y_3$
8	$\frac{dy_1}{dt} = y_2 - y_1 + y_1 y_2$ $\frac{dy_2}{dt} = y_1 - y_2 - y_1^2 - y_2^3$	16	$\frac{dy_1}{dt} = y_1^3 - y_2$ $\frac{dy_2}{dt} = y_1 + y_2^3$

3. Имеется емкость с мешалкой проточного типа. Объем емкости $V = 1000 \text{ м}^3$, расход потока $g = 100 \text{ м}^3/\text{ч}$. В момент времени, принятый за нулевой, на вход потока поступает какое-то вещество:
- а) концентрация вещества в потоке не изменяется в течение последующего времени ($c = 1.2 \text{ моль}/\text{м}^3$);
 - в) концентрация ($c = 1.2 \text{ моль}/\text{м}^3$) изменяется в виде единичного импульса в течение двух минут. Составить математическое описание и S-модель процесса и исследовать изменение концентрации вещества на выходе из емкости.

4. ВОПРОСЫ ДЛЯ САМОКОНТРОЛЯ ЗНАНИЙ

1. От какого словосочетания происходит название пакета MATLAB?
2. Что такое toolboxes MATLAB?
3. Назначение Simulink.
4. Какая информация содержится в файлах с расширением .mdl?
5. Какая информация содержится в файлах с расширением .m?
6. Какая информация содержится в файлах с расширением .mat?
7. Опишите технологию создания S – модели.
8. По какому принципу формируется библиотека Simulink?
9. Как осуществляется навигация по разделам библиотеки?
10. Каково правило соединения блоков информационными потоками?
11. Каким образом выполняется настройка параметров блоков?
12. Каково назначение блока **mux**?
13. Чем определяется число входов блока **mux**?
14. С помощью каких блоков можно сгенерировать нормально распределённый случайный сигнал?
15. С помощью какого блока можно сгенерировать равномерно распределённый случайный сигнал?
16. Назовите назначение и параметры блока **Puls Generator**.
17. Каково назначение блока **Gain**?
18. Каково назначение блока **Algebraic Constraint**?
19. Как осуществить вывод нескольких графиков на один экран **Scope**?
20. С помощью каких блоков можно получить аналитическое решение системы линейных алгебраических уравнений?
21. С помощью каких блоков можно получить решение системы обыкновенных дифференциальных уравнений (ОДУ)?
22. Как задаются начальные условия для решения ОДУ в модели?
23. Каким образом устанавливается время моделирования в Simulink?
24. Как можно управлять логикой и временем выполнения модели?

ЛАБОРАТОРНАЯ РАБОТА № 2.

ДЕТЕРМИНИРОВАННЫЕ МОДЕЛИ С НЕПРЕРЫВНЫМ ВРЕМЕНЕМ. D-СХЕМЫ

ЦЕЛЬ РАБОТЫ: освоить технологию построения и исследования непрерывно-детерминированных моделей с помощью *Simulink*.

1. ПОСТАНОВКА ЗАДАЧИ

Современное производство характеризуется тем, что некоторая часть производимой продукции (в стоимостном выражении) возвращается в виде инвестиций (т.е. части конечной продукции, используемой для создания основных фондов производства) в производство. При этом время возврата и ввода в оборот новых фондов может быть различным для различного рода производства. Необходимо промоделировать данную ситуацию и выявить динамику изменения величины основных фондов производства (капитала).

Сложность и многообразие, слабая структурированность и плохая формализуемость основных экономических механизмов, определяющих работу предприятий не позволяют преобразовать процедуры принятия решений в экономической системе в полностью эффективные математические модели и алгоритмы прогнозирования. Поэтому часто эффективно использование простых, но гибких и надёжных процедур принятия решения.

Рассмотрим одну такую простую модель. Эта модель будет полезна для прогноза событий и связанных с ними социально-экономических процессов.

2. ПОСТРОЕНИЕ МОДЕЛИ

Динамика изменения величины капитала определяется, в основном, в данной модели простыми процессами производства и описывается так называемыми обобщенными коэффициентами амортизации (расхода фондов) и потока инвестиций (часть конечного продукта, используемого в единицу времени для создания основных фондов). Эти коэффициенты - относительные величины (за единицу времени). Необходимо разработать и исследовать модель динамики основных фондов.

Пусть $x(t)$ - величина основных фондов (капитала) в момент времени t , где $0 \leq t \leq N$. Через промежуток времени Δt она будет равна $x(t + \Delta t)$. Абсолютный прирост равен $\Delta x = x(t + \Delta t) - x(t)$. Относительный прирост будет равен

$$\Delta x = [x(t + \Delta t) - x(t)] / \Delta t.$$

Примем следующие допущения:

1. поток инвестиций задается в виде известной функции $y(t)$;

2. коэффициент амортизации фондов считается неизменным и равным m . Изменение основных фондов прямо пропорционально текущей величине капитала.

Считая $\Delta t > 0$, а также учитывая определение производной, получим из предыдущего соотношения следующее математическое выражение закона изменения величины капитала - математическую модель (уравнение) динамики капитала:

$$x'(t) = y(t) - mx(t), \quad x(0) = x_0, \quad (2.1)$$

где $x(0)$ - начальное значение капитала в момент времени $t=0$.

Эта простейшая модель не отражает того факта, что социально-экономические ресурсы производства таковы, что между выделением инвестиций и их введением и использованием в выпуске новой продукции проходит некоторое время - лаг. Учитывая это, можно записать модель (2.1):

$$x'(t) = y(t-T) - mx(t), \quad x(0) = x_0 \quad (2.2)$$

3. ВЫПОЛНЕНИЕ ЗАДАНИЯ

Используя рассуждения, приведенные в пункте 1, **привести модель к канонической форме D-схемы**, сформировать S-модель в среде *Simulink* и выполнить исследование полученной модели в соответствии с нижеприведенными заданиями.

1. Основные фонды в момент времени $t=0$ были равны 50000 руб. Поток инвестиций постоянный и равен 2000 руб./сутки, $m=0.02$, $T=5$ суток. Что произойдет с капиталом через год и почему?
2. Поток инвестиций постоянный. Построить модель динамики (посуточно) и найти: величину основных фондов через t суток и минимальный капитал в течение этого времени. Какова максимально допустимая величина лага? Значения параметров, необходимые для расчётов, приведены в табл.2.1.
3. Поток инвестиций случайный (по нормальному закону). Остальные данные в табл.2.1. Найти: величину основных фондов через t суток и минимальный капитал в течение этого времени. Какова максимально допустимая величина лага?
4. Инвестиции поступают в течение 10 суток с лагом T суток в объеме 10у руб/сутки (Табл.2.1), а затем не поступают вовсе. Какова динамика капитала? Через какое время потребуются инвестиции, чтобы капитал не стал равным нулю?
5. Инвестиции поступают в течение года периодически в виде импульсов с лагом 10 суток. Определить величину инвестиций (амплитуду) и период

(при ширине импульса 50% периода), чтобы по истечении полугода капитал стал равным 210 ± 15 руб. Начальный капитал 1000руб., $m = 0.025$.

6. Поток инвестиций убывает во времени по экспоненте $10 \cdot \exp(-0.001 \cdot t)$. Начальный капитал и коэффициент амортизации в табл.2.1. Через какое время капитал предприятия будет нулевым?
7. Поток инвестиций возрастает по линейному закону $y = a \cdot t$ (блок Ramp в источниках). Начальный капитал и коэффициент амортизации в табл.2.1. Найти такое значение коэффициента a , чтобы через год капитал предприятия стал равным начальному.
8. Поток инвестиций случайный, имеет нормальный закон распределения. Начальный капитал и коэффициент амортизации в табл.2.1. Найти математическое ожидание потока такое, чтобы по истечении 100 дней величина капитала была близка к первоначальной и далее поддерживалась на этом уровне. (Дисперсия не должна превышать 8-10 % от математического ожидания)

Таблица 2.1

Номер варианта	y , руб./сутки	x_0 , руб.	m , 1/сутки	T , сутки	t , сутки
1	111	1000	0.025	5	50
2	125	900	0.030	10	55
3	115	980	0.035	6	45
4	98	1100	0.020	9	60
5	130	950	0.017	8	53
6	150	840	0.021	6.5	48
7	90	1200	0.033	5.8	57
8	100	970	0.018	7	40
9	120	890	0.023	5.5	35
10	135	1050	0.019	12	38
11	140	1150	0.024	11	60
12	145	975	0.028	11.5	58
13	95	1030	0.034	7.5	52
14	113	986	0.038	8.5	54
15	148	1080	0.036	9.5	56
16	137	1060	0.027	10.5	59

4. ВОПРОСЫ ДЛЯ САМОКОНТРОЛЯ ЗНАНИЙ

1. Что означает понятие D-схема?
2. Почему процесс изменения капитала можно представить непрерывно-детерминированной моделью?
3. Запишите модель изменения величины капитала во времени.

4. Каковы причины изменения величины капитала?
5. Какие допущения приняты при разработке модели?
6. Что такое лаг?
7. Каким образом можно представить модель изменения величины капитала во времени в канонической форме D-схемы?
8. Каким образом могут поступать инвестиции в решённых Вами задачах?
9. Какие блоки моделируют поток инвестиций в Simulink моделях?
10. В каком случае величина капитала будет постоянной, даже если инвестиции поступают?
11. Какими параметрами задаётся случайное поступление инвестиций в соответствии с нормальным распределением?
12. В каких единицах измерения устанавливается время моделирования?
13. Что является начальным условием при моделировании той или иной ситуации?
14. Как осуществляется управление логикой и временем выполнения модели?

ЛАБОРАТОРНАЯ РАБОТА №3.

МАТЕМАТИЧЕСКОЕ ПЛАНИРОВАНИЕ ЭКСПЕРИМЕНТА НА МАТЕМАТИЧЕСКОЙ МОДЕЛИ СИСТЕМЫ

ЦЕЛЬ РАБОТЫ: На примере S-модели системы терморегулирования дома освоить принципы и методы математического планирования эксперимента на моделях. Найти регрессию, позволяющую прогнозировать стоимость обогрева дома в течение определённого периода времени в зависимости от желаемой температуры в доме и температуры на улице.

1. ОПИСАНИЕ МОДЕЛИ

S-модель системы терморегулирования отдельного дома, представленная на рис.3.1, входит в состав демонстрационных моделей Simulink. Файл **thermo.mdl**.

Температура в доме контролируется с помощью датчика температуры (сигнал **Tin** на выходе из подсистемы **House**). Сигнал датчика сравнивается с опорным сигналом температуры (блок **Set Point**), и разность (сигнал после сумматора) используется для импульсного управления нагревателем (подсистема **Thermostat**). Релейный блок **Relay** в этой подсистеме работает следующим образом: если сигнал на входе меньше некоторого порога, то на выходе получается сигнал одного уровня (обычно 0), а если порог превышен, то сигнал на выходе становится другого уровня (обычно 1). Порог и уровни сигнала устанавливаются в окне параметров блока. Если сигнал на выходе подсистемы **Thermostat** принимает значение 1, включается воздуходувка – блок **Heater Blower**. (Обогрев дома осуществляется нагретым до 30⁰С воздухом.)

В подсистеме **House** определяется температура в доме (**Tin**), при этом учитывается количество тепла, поступающего с горячим воздухом (**heater QDot**) и потери тепла через стены и окна. Величина потерь обратно пропорциональна эквивалентному термическому сопротивлению для всего здания (**Req**) и прямо пропорциональна разности температур на улице и в доме. Скорость изменения температуры определяется делением суммарного количества тепла на теплосодержание (**Mc**) общей массы воздуха (**M**) в доме. В результате интегрирования определяется температура в данный момент времени. Изменение температуры на улице в течение суток задается в блоке **Daily Temp Variation** в виде синусоидального сигнала. Разность температур днем и ночью определяется амплитудой синусоиды. В системе предусмотрен контроль расходов на обогрев дома в долларах – блок **Dollar Gain**. Период времени, за который определяются расходы, задаётся временем моделирования в секундах.

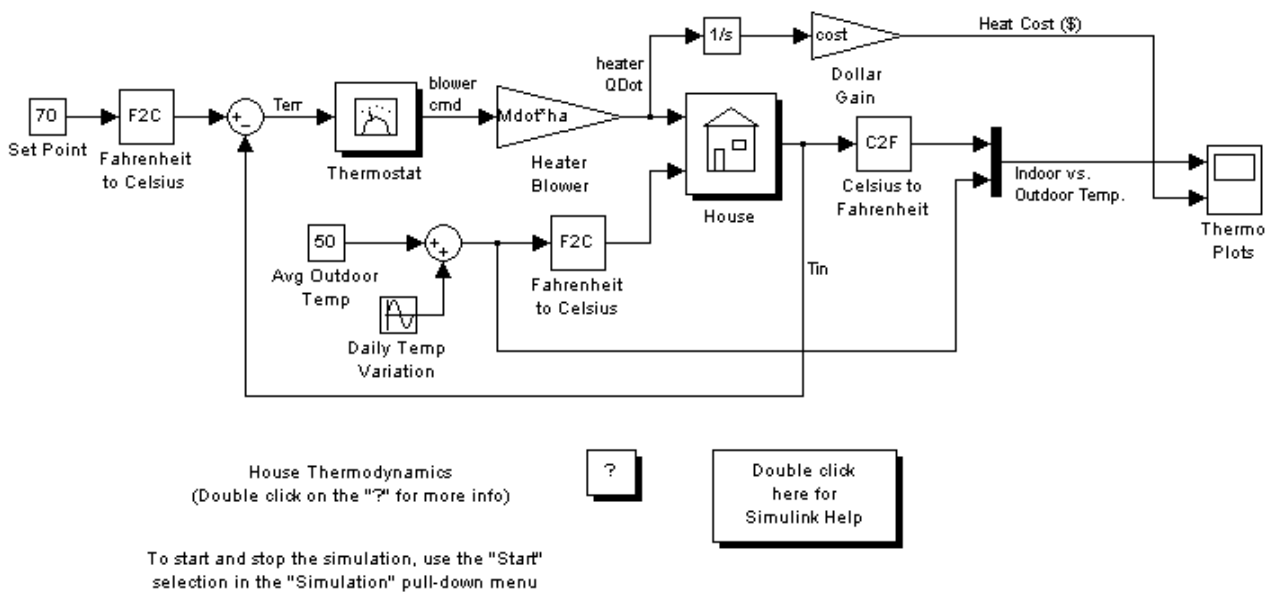


Рис.3.1. S-модель системы терморегулирования дома

Параметры большинства блоков, теплофизические и термодинамические характеристики воздуха и материалов, из которых построен дом, а также размеры дома задаются в m – файле **thermdat.m**, содержащем подробные комментарии к данным. В случае необходимости данные могут быть изменены.

2. ПЛАНИРОВАНИЕ ЭКСПЕРИМЕНТА

При планировании эксперимента варьируемые переменные называются факторами, измеряемая переменная, значения которой зависят от факторов – откликом. В соответствии с целью работы **факторы - это опорная температура в доме и температура на улице, отклик - стоимость обогрева дома.** Прежде чем перейти к составлению плана, рассмотрим терминологию, принятую в планировании эксперимента.

Уровни факторов

Для выполнения эксперимента необходимо определить область экспериментирования. С этой целью экспериментатор задаёт диапазон значений для каждого фактора. В планировании эксперимента значения факторов принято называть уровнями:

- нижний уровень фактора – это его минимальное значение (Z^{\min});
- верхний уровень фактора – это его максимальное значение (Z^{\max});
- основной уровень фактора – это его среднее значение (Z^0).

Например, в эксперименте варьируется температура (Z_1) в интервале от 100 до 200⁰C. Для этого фактора нижний уровень: 100⁰C, верхний уровень: 200⁰C, основной уровень: 150⁰C.

Центр плана

Опыт, значения факторов в котором соответствуют основному уровню, называется центром плана. Таким образом, координата центра плана по i -му фактору:

$$Z_i^0 = (Z_i^{\max} + Z_i^{\min})/2 \quad i = 1, 2..n, \quad (3.1)$$

где n – число факторов в эксперименте.

Интервал варьирования

Интервал варьирования (ΔZ_i) – это разность между верхним и основным уровнем фактора или разность между основным и нижним уровнями. Соответственно разность между верхним и нижним уровнями составляет удвоенный интервал варьирования, поэтому:

$$\Delta Z_i = (Z_{i\max} - Z_{i\min})/2 \quad i = 1, 2..n \quad (3.2)$$

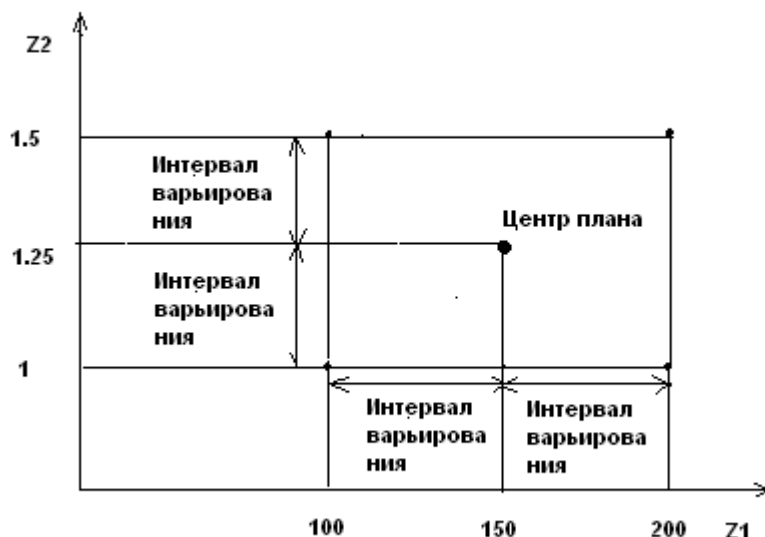


Рис.3.2. Координаты центра плана и интервалы варьирования

Например, в эксперименте варьируется температура (Z_1) в диапазоне от 100 до 200⁰С и концентрация (Z_2) в диапазоне от 1 до 1.5 моль/л. Интервалы варьирования в соответствии с (2): 50⁰С и 0.25 моль/л. Координаты центра плана и интервалы варьирования показаны на рис.3.2.

Кодированные факторы

Математический план эксперимента не должен зависеть от физической природы факторов, вид плана определяется только числом факторов и характером идентифицируемой регрессии. Поэтому все планы разработаны для кодированных факторов (X), т.е. факторов, приведённых к единой системе координат. Переход от фактора Z_i , значения которого имеют определённую размер-

ность, будем называть такие факторы размерными, к кодированному фактору X_i , будем называть такие факторы безразмерными, осуществляется по формуле:

$$X_i = (Z_i - Z_i^0) / \Delta Z_i \quad (3.3)$$

Если в эту формулу подставить соответствующие выражения (3.1) и (3.2), то получим

$$\begin{array}{ll} \text{для } Z_i = Z_i^{\max} & - X_i = 1 \\ \text{для } Z_i = Z_i^{\min} & - X_i = -1 \\ \text{для } Z_i = Z_i^0 & - X_i = 0 \end{array}$$

Теперь перейдем непосредственно к составлению плана эксперимента, в соответствии с которым варьируются факторы и измеряется отклик. Зависимость отклика от факторов – регрессия. Вид регрессии зависит от типа плана. Для получения линейной регрессии будем использовать ПФЭ (полный факторный эксперимент) типа 2^n . Если линейная регрессия окажется неадекватной, следует дополнить этот план до центрального композиционного плана (ЦКП) 2-го порядка, например, ортогонального (ОЦКП).

2.1. План полного факторного эксперимента

План ПФЭ включает все возможные комбинации из n факторов, каждый из которых принимает 2 уровня (нижний и верхний). Число опытов m равно числу возможных комбинаций:

$$m = 2^n \quad (3.4)$$

Для $n = 2$ число опытов $m = 4$ и построение плана можно наглядно показать на рис. 3.3. Координаты вершин квадрата (1-4) соответствуют условиям соответствующих опытов: (-1,1), (1,1), (-1,-1), (1,-1)

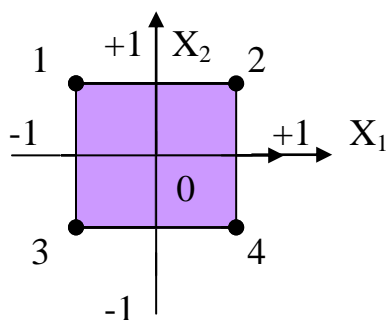


Рис. 3.3. ПФЭ для двух факторов

При $n = 3$ число опытов $m = 8$, условия опытов соответствуют координатам вершин куба, и матрица планирования имеет вид:

Матрица планирования

Номер опыта	X_1	X_2	X_3	Y
1	-1	-1	-1	Y_1
2	+1	-1	-1	Y_2
3	-1	+1	-1	Y_3
4	+1	+1	-1	Y_4
5	-1	-1	+1	Y_5
6	+1	-1	+1	Y_6
7	-1	+1	+1	Y_7
8	+1	+1	+1	Y_8

Матрица планирования ПФЭ для любого числа факторов обладает следующими свойствами:

1. $\sum_{j=1}^m X_{j,u} \cdot X_{j,i} = 0;$ $u \neq i; u, i = 1, 2, \dots, n;$
2. $\sum_{j=1}^m X_{j,i} = 0;$ $i = 1, 2, \dots, n;$
3. $\sum_{j=1}^m X_{j,i}^2 = m;$ $i = 1, 2, \dots, n;$

Первое свойство – равенство нулю скалярных произведений двух любых векторов-столбцов – называется свойством ортогональности матрицы планирования. Второе свойство – равенство нулю суммы элементов любого вектора-столбца – называется свойством симметричности матрицы планирования.

Для выполнения эксперимента необходимо от кодированных факторов перейти к размерным:

$$Z_i = Z_i^0 + \Delta Z_i \cdot X_i \quad (3.5)$$

Результаты измерений отклика во всех опытах составляют вектор наблюдений (столбец Y табл.1).

Регрессионный анализ результатов ПФЭ за счёт свойств матрицы планирования существенно упрощается. Зависимость отклика от факторов предполагается линейной

$$\hat{Y} = b_0 + b_1 X_1 + b_2 X_2 + \dots + b_n X_n \quad (3.6)$$

Например, для $n=3$:

$$\hat{Y} = b_0 + b_1 X_1 + b_2 X_2 + b_3 X_3 \quad (3.7)$$

Коэффициенты регрессии b_i (линейные эффекты) определяются методом наименьших квадратов:

$$B = (X^T X)^{-1} \cdot X^T Y,$$

X – матрица факторов, число строк в которой соответствует числу опытов m , а число столбцов $n+1$ (число факторов плюс один фиктивный фактор, который вводится для оценки коэффициента b_0 и во всех опытах равен 1);

X^T - транспонированная матрица факторов;

Y - вектор значений отклика в m опытах.

Благодаря свойству ортогональности информационная матрица становится диагональной:

$$X^T X = \begin{bmatrix} m & 0 & \dots & 0 \\ 0 & m & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & m \end{bmatrix},$$

матрица, обратная информационной, также диагональная

$$(X^T X)^{-1} = \begin{bmatrix} 1/m & 0 & \dots & 0 \\ 0 & 1/m & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & 1/m \end{bmatrix},$$

и коэффициенты регрессии рассчитываются по следующим формулам:

$$b_0 = \frac{1}{m} \sum_{j=1}^m y_j; \quad b_i = \frac{\sum_{j=1}^m x_{ji} y_j}{m}, \quad i = 1, 2, \dots, n \quad (3.8)$$

где y_j - экспериментальное значение отклика в j -м опыте.

Адекватность регрессии проверяют по коэффициенту детерминации (R^2), который показывает насколько модель регрессии лучше модели среднего:

$$R^2 = 1 - \frac{SSE}{SST}, \quad (3.9)$$

SSE - сумма квадратов, определяющая рассеяние экспериментальных значений отклика относительно регрессии

$$SSE = \sum_{i=1}^m (\hat{y}_i - y_i)^2,$$

SST - сумма квадратов, определяющая рассеяние измерений отклика относительно среднего по всем опытам (\bar{y})

$$SST = \sum_{i=1}^m (\bar{y} - y_i)^2, \quad \text{где} \quad \bar{y} = \sum_{i=1}^m \frac{y_i}{m}.$$

Если $SSE \approx SST$, то модель регрессии представляет данные примерно с той же точностью, что и просто среднее и использовать регрессию не имеет смысла. В этом случае $\frac{SSE}{SST} \approx 1$ и $R^2 \approx 0$.

Если $SSE \ll SST$, то регрессия представляет данные значительно точнее, чем просто среднее и имеет смысл использовать регрессию. При этом $\frac{SSE}{SST} \approx 0$ и $R^2 \approx 1$. Таким образом, чем ближе значение R^2 к 1, тем с большей уверенностью можно утверждать, что регрессия адекватна. Коэффициент детерминации интерпретируют, как долю изменчивости отклика от факторов, объясняемую регрессией. Например, $R^2 = 0.8$ означает, что 80% изменчивости Y от X объясняется регрессией, а 20% - случайностью. В идеале желательно иметь объяснение если не для всей, то хотя бы для большей части исходной изменчивости.

Более строго предположение о значимости доли изменчивости, объясняемой регрессией, можно проверить по критерию F:

$$F = \frac{SSR(m - n - 1)}{(SSE)n},$$

SSR - сумма квадратов, определяющая рассеяние среднего экспериментального значения отклика относительно регрессии

$$SSR = \sum_{i=1}^m (\hat{y}_i - \bar{y})^2.$$

Очевидно, что чем больше F, тем лучше регрессия описывает экспериментальные данные по сравнению со средним. Учитывая, что

$$SST = SSR + SSE,$$

F можно выразить через коэффициент детерминации:

$$F = \frac{R^2(m - n - 1)}{n(1 - R^2)}. \quad (3.10)$$

Для рассчитанного значения F по распределению Фишера можно вычислить уровень значимости **p-level**. Если **p-level** < 0.05, то с вероятностью 0.95 можно признать долю изменчивости, объясняемую регрессией значимой и модель адекватной. В противном случае – изменчивость объясняется случайностью и линейная регрессионная модель неприемлема. Однако результаты ПФЭ позволяют идентифицировать регрессионную модель, которая кроме линейных эффектов включает эффекты взаимодействия (произведения) факторов:

$$\hat{Y} = b_0 + b_1 X_1 + b_2 X_2 + b_3 X_3 + b_{12} X_1 X_2 + b_{13} X_1 X_3 + b_{23} X_2 X_3. \quad (3.11)$$

Для расчёта коэффициентов **b₁₂**, **b₁₃**, **b₂₃** при соответствующих эффектах взаимодействия необходимо матрицу плана, приведённую в табл.1, дополнить столбцами произведений – табл.2.

Таблица 3.2

Матрица планирования

Номер опыта	X ₁	X ₂	X ₃	X ₁ X ₂	X ₁ X ₃	X ₂ X ₃	Y
1	-1	-1	-1	+1	+1	+1	Y ₁
2	+1	-1	-1	-1	-1	+1	Y ₂
3	-1	+1	-1	-1	+1	-1	Y ₃
4	+1	+1	-1	+1	-1	-1	Y ₄
5	-1	-1	+1	+1	-1	-1	Y ₅
6	+1	-1	+1	-1	+1	-1	Y ₆
7	-1	+1	+1	-1	-1	+1	Y ₇
8	+1	+1	+1	+1	+1	+1	Y ₈

Дополнительные коэффициенты рассчитываются по формуле (3.8). Адекватность регрессии с эффектами взаимодействия проверяется так же, как и адекватность линейной модели. Если и эта модель окажется неадекватной, следует расширить план ПФЭ до плана второго порядка, который позволит идентифицировать квадратичную регрессию.

2.2. Композиционный план 2-го порядка

Композиционный план 2-го порядка для **n** факторов составляется следующим образом:

- в качестве ядра плана используется ПФЭ;
- добавляется $2n$ звёздных точек, расположенных на осях факторов на расстоянии звёздного плеча $\pm\alpha$ от центра плана;
- добавляется m_0 опытов в центре плана.

Отсюда общее число опытов в плане:

$$m = 2^n + 2n + m_0. \quad (3.12)$$

Величина звёздного плеча α зависит от типа композиционного плана (ортогональный или ротатабельный), числа факторов и от числа опытов в центре плана. Так в ортогональном композиционном плане 2-го порядка для 3-х факторов с числом опытов в центре плана 2, величина звёздного плеча составляет 1,287. Матрица этого плана представлена в табл.3.

Таблица 3.3

Матрица планирования

Номер опыта	X1	X2	X3
1	-1	-1	-1
2	+1	-1	-1
3	-1	+1	-1
4	+1	+1	-1
5	-1	-1	+1
6	+1	-1	+1
7	-1	+1	+1
8	+1	+1	+1
9	-1,287	0	0
10	+1,287	0	0
11	0	-1,287	0
12	0	+1,287	0
13	0	0	-1,287
14	0	0	+1,287
15	0	0	0
16	0	0	0

Обработка результатов планирования позволяет идентифицировать полную квадратичную регрессию:

$$\hat{Y} = b_0 + b_1 X_1 + b_2 X_2 + b_3 X_3 + b_{12} X_1 X_2 + b_{13} X_1 X_3 + b_{23} X_2 X_3 + b_{11} X_1^2 + b_{22} X_2^2 + b_{33} X_3^2. \quad (3.13)$$

Адекватность регрессии проверяется по коэффициенту детерминации, как было показано выше.

3. ВЫПОЛНЕНИЕ ПОСТАВЛЕННОЙ ЗАДАЧИ

1. Составить подробное математическое описание исследуемой задачи.
2. S-модель оформить в виде подсистемы с двумя входами и одним выходом.
3. В основной S-модели организовать вывод вектора откликов в рабочее пространство. Для наглядности и контроля процесса экспериментирования на модели организовать вывод матрицы планирования факторов и вектор отклика на 2 дисплея.
4. Составить m-файл сценария, в котором генерируется план эксперимента заданного типа в кодированных факторах, выполняется пересчет факторов в размерные, осуществляется обращение к S-модели для выполнения эксперимента, считываются значения отклика из рабочего пространства и результаты эксперимента обрабатываются регрессией соответствующего вида.
5. Запустить сценарий, предварительно выполнив программу **thermdat.m**
6. Необходимые файлы находятся в папке Y:\230201_ФММ\ЛАБОРАТОРНЫЕ\КЛАБ3

3.1. Реализация m-сценариев для выполнения задачи

В нашем случае мы имеем два фактора: опорная температура в доме и температура на улице. Откликом будет стоимость обогрева дома.

Линейная регрессия в кодированных факторах:

$$\hat{Y} = b_0 + b_1 X_1 + b_2 X_2 \quad (3.14),$$

в размерных факторах:

$$\hat{Y} = a_0 + a_1 Z_1 + a_2 Z_2, \quad (3.15)$$

Z_1 - опорная температура в доме в градусах Цельсия или Фаренгейта;

Z_2 - температура на улице в градусах Цельсия или Фаренгейта.

Для получения линейной регрессии достаточно составить ПФЭ типа 2^2 :

N	X_0	X_1	X_2	Z_1	Z_2
1	1	-1	-1	60	30
2	1	+1	-1	80	30
3	1	-1	+1	60	60
4	1	+1	+1	80	60

Z_i находим по формуле (5), при условии: $Z_1^0 = 70F$, $\Delta Z_1 = 10F$, $Z_2^0 = 45F$, $\Delta Z_2 = 15F$.

Для формирования плана и обработки результатов эксперимента будем использовать функции из пакета расширения (toolbox) MATLAB, который называется

stats. Можно получить коэффициенты регрессии (3.14) или (3.15) в зависимости от используемой в расчётах матрицы факторов.

M-сценарий для идентификации линейной модели будет выглядеть следующим образом:

% - вектор столбец фиктивного фактора

X1 = [1;1;1;1];

% – формирование плана

X2 = CORDEXCH (2,4,'linear'); *% функция toolbox stats- формирует план*

% ПФЭ как двумерный массив X2;

% список аргументов функции: число факторов(2), число опытов(4),

% 'тип регрессии'

% Переход к размерным факторам

for k = 1:4

z1(k) = 70+10*X2(k,1) ; *% точка с запятой используется для подавления*

% вывода значения в командное окно

z2(k) = 45+15*X2(k,2);

end;

% выполнение эксперимента

% на модели

% - загрузка S-модели

load_system('thermo');

%-выполнение

sim('thermo');

%-считываем значения отклика из рабочего пространства

y1 = simout(end,1:end);

% - формируем столбец значений отклика

y = y1' ; *% апостроф означает транспонирование*

% - матрица размерных факторов

X = [X1,z1',z2']

%REGRESS- функция toolbox stats, выполняет регрессионный анализ

% результатов эксперимента;

% список аргументов: y – вектор значений отклика, X – матрица факторов

% 0.05 – уровень значимости, назначенный при проверке адекватности модели

[B, BINT,R,RINT, STATS] = REGRESS (y,X,0.05)

% [B] – столбец коэффициентов регрессии

% [BINT] – доверительные интервалы для коэффициентов

% [R] – столбец остатков

% [RINT] – доверительные интервалы для остатков

% [STATS] – выдает три значения: коэффициент детерминации, критерий Фишера, расчётный уровень значимости (если расчётный уровень значимости больше назначенного, то модель признается не адекватной)
yr = X*B *% расчетные значения функции отклика.*

Если линейная регрессия окажется неадекватной, следует идентифицировать регрессию с эффектами взаимодействия.

$$\hat{Y} = b_0 + b_1 X_1 + b_2 X_2 + b_{12} X_1 X_2 \quad (3.16),$$

в размерных факторах:

$$\hat{Y} = a_0 + a_1 Z_1 + a_2 Z_2 + a_{12} Z_1 Z_2 \quad (3.17)$$

Выполните необходимые для этого изменения в m - файле самостоятельно.

Если и эта модель окажется неадекватной, необходимо дополнить план ПФЭ до ортогонального композиционного плана 2-го порядка.

Уравнение квадратичной регрессии, идентифицированной по результатам эксперимента, выполненного в соответствии с композиционным планом, будет иметь следующий вид:

$$\hat{Y} = b_0 + b_1 X_1 + b_2 X_2 + b_{12} X_1 X_2 + b_{11} X_1^2 + b_{22} X_2^2, \quad (3.18)$$

в размерных факторах:

$$\hat{Y} = a_0 + a_1 Z_1 + a_2 Z_2 + a_{12} Z_1 Z_2 + a_{11} Z_1^2 + a_{22} Z_2^2. \quad (3.19)$$

Число опытов в плане, определённое по формуле (9):

$$m = 2^2 + 2*2+1 = 9.$$

Запишем изменения в m – файл:

Вместо **'linear'** используем **'quadratic': CORDEXCH (2,9,'quadratic')**

С учетом внесенных изменений, m-сценарий будет иметь вид:

% - вектор столбец фиктивного фактора

X1 = [1;1;1;1;1;1;1;1;1];

% – формирование плана

X2 = CORDEXCH (2,9,'quadratic');

% Переход к размерной форме

% Переход к размерным факторам

for k = 1:9

z1(k) = 70+10*X2(k,1) ;

```

z2(k) = 45+15*X2(k,2);
% расчет значений эффекта взаимодействия
z3(k) = z1(k)*z2(k);
% расчет значений квадратичных эффектов
z4(k) = z1(k)*z1(k);
z5(k) = z2(k)*z2(k);
end;

% - загрузка
load_system('thermo');

%-выполнение
sim('thermo');

% - формируем столбец откликов
y1 = simout(end,1:end);
y = y1' ;

% - матрица размерных факторов
z = [X1, z1', z2', z3', z4', z5']

%REGRESS функция
[B, BINT,R,RINT, STATS] = REGRESS (y,z,0.05)
yr = z*B % расчетные значения функции отклика

```

4. ВАРИАНТЫ ЗАДАНИЙ ДЛЯ ВЫПОЛНЕНИЯ ЛАБОРАТОРНОЙ РАБОТЫ

Номер варианта	Центр плана		Размеры дома, м			Окна			Тёплый воздух для обогрева дома	
	Z ₁ ⁰ , °C	Z ₂ ⁰ , °C	L	W	H	N шт	H м	W м	T °C	ск-ть, кг/с
1	21	-1	30	10	4	6	1	1	30	0,10
2	20	0	29	8	3,5	5	1,5	1,5	32	0,20
3	15	2	28	9	3	4	1,5	2	35	0,12
4	18	4	30	10	3,8	5	1,2	1,5	36	0,15
5	16	5	32	12	5	6	1	1,2	31	0,18
6	17	6	31	11	4	6	1	1,4	32	0,16
7	19	7	33	11	4,5	6	1	1,5	30	0,14
8	20	8	35	13	5	6	1	1,6	34	0,16
9	15	9	34	12	4,8	6	1	1,5	35	0,15
10	21	10	25	7	3	4	2	2	36	0,10

Номер варианта	Центр плана		Размеры дома, м			Окна			Тёплый воздух для обогрева дома	
	$Z_1^0, ^\circ\text{C}$	$Z_2^0, ^\circ\text{C}$	L	W	H	N шт	H м	W м	T $^\circ\text{C}$	ск-ть, кг/с
11	18	7	26	7	3	5	1,8	1,8	38	0,20
12	20	5	27	8	3	5	2	1,5	31	0,15
13	22	0	30	9	3,5	6	1,5	2	32	0,18
14	18	-1	25	8	3,2	4	1,5	1,8	30	0,11
15	19	4	25	9	3,8	5	1,8	2	35	0,13
16	16	10	27	10	3,2	5	2	1,5	34	0,14
17	20	2	32	11	4	6	1	1,2	33	0,19
18	21	6	34	13	5	6	1	1,3	32	0,12

Принять: $\Delta Z_1 = 10F$, $\Delta Z_2 = 15F$.

Значения остальных параметров модели дома оставить без изменения.

Формулы перевода температуры из Фаренгейта в Цельсия:

$$C = 5/9 \cdot (F - 32)$$

из Цельсия в Фаренгейта:

$$F = 9/5 \cdot C + 32$$

5. ВОПРОСЫ ДЛЯ САМОКОНТРОЛЯ ЗНАНИЙ

1. С какой целью используется математический план эксперимента?
2. С какой целью используется математический план эксперимента в лабораторной работе?
3. Каким образом задаётся область экспериментирования?
4. Что такое уровни факторов?
5. Число и названия факторов в лабораторной работе.
6. Что является откликом в лабораторной работе?
7. Как определяется число опытов в ПФЭ?
8. Свойства плана ПФЭ типа 2^n .
9. Формула связи кодированных и размерных факторов.
10. Назовите этапы регрессионного анализа результатов планированного эксперимента.
11. Вид регрессии, которая может быть получена с помощью ПФЭ типа 2^n .
12. Каким методом определяются коэффициенты регрессии?
13. Какова размерность матрицы факторов, информационной матрицы и матрицы, обратной информационной в лабораторной работе?

14. Как проверяется адекватность регрессионной модели?
15. Каким образом конструируется центральный композиционный план второго порядка?
16. Вид регрессии, которая может быть получена с помощью центрального композиционного плана второго порядка.
17. Число и координаты звёздных точек.
18. От чего зависит величина звёздного плеча?
19. Состав и назначение программных модулей, используемых в лабораторной работе.
20. Какие пакеты расширения MATLAB используются в лабораторной работе?
21. Для каких целей может быть использована полученная в лабораторной работе модель?
22. В каком месте программы обработки результатов планированного эксперимента, и каким образом определяется форма регрессии (в размерных или кодированных факторах)?
23. В какой форме - в размерных или кодированных факторах регрессия наиболее удобна для практического использования?

ЛАБОРАТОРНАЯ РАБОТА №4.

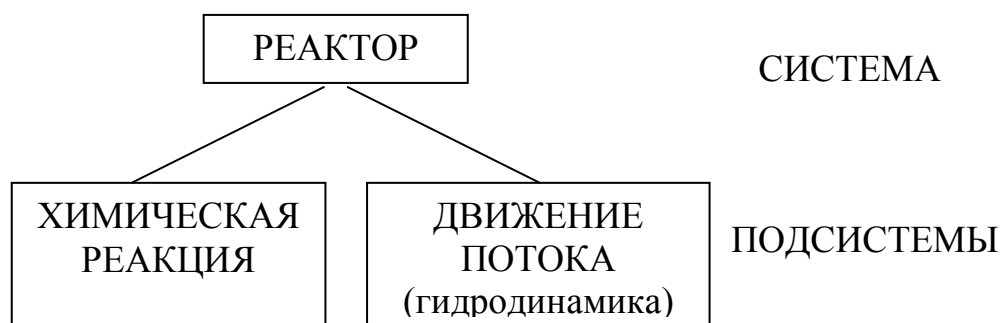
МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ ХИМИЧЕСКИХ ПРОЦЕССОВ.

МОДЕЛИ РИВ и РИС

Цель работы: На основе принципов системного анализа составить математическое описание гомогенных реакторов: идеального вытеснения, идеального смешения и каскада реакторов идеального смешения, в которых протекает сложная химическая реакция с заданным механизмом. Выполнить программную реализацию в виде S – моделей и по результатам моделирования рекомендовать наиболее эффективный реактор.

1. СИСТЕМНЫЙ АНАЛИЗ ГОМОГЕННОГО ХИМИЧЕСКОГО РЕАКТОРА

Цель системного анализа (СА) в данном случае - разработка математической модели реактора. Реактор осуществляет преобразование исходных реагентов на входе в продукты химической реакции на выходе. В гомогенном реакторе все компоненты находятся в одной фазе (жидкость, газ). Реактор можно представить как систему, элементами которой являются отдельные физико-химические процессы, т.е. как физико-химическую систему (ФХС). Системный анализ, как метод исследования сложных систем, предполагает последовательный переход от общего к частному, исследование частного и объединение в целое. Таким образом, СА представляет собой сочетание двух процедур: **декомпозиции и агрегирования**. Выполним **декомпозицию (анализ) ФХС**:



Каждую из подсистем рассмотрим как систему более низкого уровня по отношению к системе «РЕАКТОР».

1.1. Математическое описание подсистемы «химическая реакция»

Подсистему «химическая реакция» можно рассмотреть как систему, в которой происходят взаимодействия между компонентами на уровне молекул. Элементы системы – компоненты, участвующие в реакции, а связь между компонентами определяется маршрутом химической реакции. Присутствуют все признаки системы (покажите самостоятельно!). Количественной характеристи-

кой элементов системы или параметрами системы являются концентрации компонентов, а параметрами, характеризующими связи – константы скорости реакций. Изменение концентраций реагентов в ходе реакции определяется кинетикой химической реакции. Если механизм химической реакции известен, то математическое описание (МО) кинетики можно представить в матричной форме. Процедура составления МО в матричной форме достаточно хорошо формализована и состоит в следующем.

1. Составляются **уравнения скоростей отдельных стадий химической реакции**:

$$r_j = k_j \cdot \prod_{i=1}^n C_i^{a_{i,j}} \quad j = 1, \dots, m, \quad (4.1)$$

где r_j – скорость j -ой стадии (число стадий m), k_j – константа скорости j -й стадии, C_i – концентрация i -го исходного реагента, $a_{i,j}$ – стехиометрический коэффициент i -го компонента в j -й стадии, n – число реагентов.

2. Составляются **уравнения скоростей реакций по компонентам**:

$$W_i = \sum_{j=1}^m \pm a_{i,j} r_j \quad i=1, \dots, n, \quad (4.2)$$

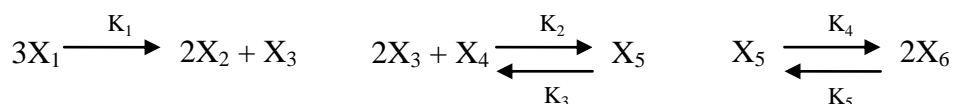
где W_i – скорость реакции по i -му компоненту.

Систему (2) с учётом (1) можно представить в матричной форме:

$$[W_i] = [a_{i,j}] [r_j] \quad (4.3)$$

где $[W_i]$ – вектор из n элементов, $[r_j]$ – вектор из m элементов, $[a_{i,j}]$ – матрица стехиометрических коэффициентов размерностью $n \times m$. Элемент матрицы, стоящий на пересечении i -й строки и j -го столбца – это стехиометрический коэффициент i -го компонента в j -й стадии взятый со знаком «-» для исходных реагентов и со знаком «+» для продуктов реакции.

Пример. Задан механизм сложной химической реакции



Число реагентов $n = 6$, число стадий $m = 5$ (по числу констант). Скорости стадий в соответствии с (1):

$$r_1 = K_1 C_{X_1}^3 \quad r_2 = K_2 C_{X_3}^2 C_{X_4} \quad r_3 = K_3 C_{X_5}$$

$$r_4 = K_4 C_{X5} \quad r_5 = K_5 C_{X6}^2$$

Скорости реакций по компонентам в соответствии с (4.3):

$$\begin{bmatrix} W_1 \\ W_2 \\ W_3 \\ W_4 \\ W_5 \\ W_6 \end{bmatrix} = \begin{bmatrix} -3 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 \\ 1 & -2 & 2 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & 1 & -1 & -1 & 1 \\ 0 & 0 & 0 & 2 & -2 \end{bmatrix} \cdot \begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \\ r_5 \end{bmatrix} \quad (4.4)$$

Система (4.4) представляет собой математическое описание кинетики химической реакции заданного механизма.

1.2. Математическое описание подсистемы «движение потока»

Подсистему «движение потока» или «гидродинамика» можно рассмотреть как систему, в которой происходит движение частиц потока жидкости или газа, причем частицы имеют различную скорость и траекторию движения, изменяющиеся случайным образом. Частицы потока – это элементы системы. Здесь также присутствуют все признаки системы (покажите самостоятельно!). Разработка математического описания этой системы встречает на своем пути определённые трудности. Невозможно описать движение каждой частицы потока, при условии, что таких частиц бесконечное множество и траектория их движения может изменяться случайным образом, поэтому в теории математического моделирования гидродинамики используется следующий подход.

Каждой частице потока ставится в соответствие время её пребывания в аппарате. Величина времени пребывания рассматривается как случайная, а доля частиц с заданным временем пребывания в общей массе частиц представляет собой вероятность случайной величины. Для получения эмпирической плотности распределения или эмпирической функции распределения времени пребывания проводят специальный эксперимент. В поток на входе в аппарат вводят некоторый индикатор в виде импульсного или ступенчатого возмущения, т.е. помечают поступающие в аппарат частицы потока, а на выходе из аппарата снимают кривую отклика в виде зависимости концентрации индикатора от времени, т.е. фиксируют выходящие из аппарата частицы. Кстати, S-модель подобного эксперимента, проведённого на ёмкости с мешалкой, рассматривалась в последней задаче лаб. раб. №1. Преобразованная определённым образом кривая отклика на импульсное возмущение представляет собой эмпирическую плотность распределения, а кривая отклика на ступенчатое возмущение – эмпирическую функцию распределения элементов потока по времени пребывания в аппарате. По виду функции или плотности распределения выбирают одну из

типовых моделей гидродинамики и при необходимости определяют её параметры.

Типовые модели гидродинамики построены на основе тех или иных допущений относительно характера движения элементов потока в аппарате и представляют собой некоторую идеализацию реального процесса. **Модель идеального вытеснения** предполагает, что все элементы потока движутся с одинаковой скоростью и имеют одинаковое время пребывания. МО движения такого потока, полученное из материального баланса по индикатору для элементарного участка аппарата длиной dx , имеет следующий вид.

$$\frac{\partial c}{\partial t} = -U \cdot \frac{\partial c}{\partial x}, \quad (4.5)$$

где c – концентрация индикатора, t – текущее время, x – координата по длине реактора, U – линейная скорость потока.

Модель идеального смешения предполагает, что поток, поступающий в аппарат, мгновенно перемешивается, вследствие чего концентрация индикатора во всём объёме аппарата одинакова и равна концентрации на выходе. МО движения такого потока, полученное из материального баланса по индикатору для аппарата в целом, имеет следующий вид.

$$\frac{dc}{dt} = \frac{1}{\bar{\tau}} \cdot (c_{\text{вх}} - c), \quad (4.6)$$

где $\bar{\tau}$ – среднее время пребывания потока в аппарате, определяемое как отношение объёма аппарата к объёмной скорости потока; $c_{\text{вх}}$ – концентрация индикатора на входе в аппарат.

Следует отметить, что модели гидродинамики (4.5) и (4.6) являются динамическими, т.к. описывают нестационарный процесс изменения концентрации индикатора при нанесении возмущения.

1.3. Математическое описание гомогенного реактора

МО реактора в целом строится на основе агрегирования (синтеза) МО подсистем. Для разработки **МО реактора идеального вытеснения (РИВ)** необходимо на основе модели гидродинамики (4.5) составить материальный баланс по всем компонентам, участвующим в химической реакции с учётом кинетических закономерностей (4.3).

$$\frac{\partial c_i}{\partial t} = -U \cdot \frac{\partial c_i}{\partial x} + W_i \quad i = 1, 2, \dots, n \quad (4.7)$$

В стационарных условиях работы реактора $\frac{\partial c_i}{\partial t} = 0$. С учётом того, что $dx/U = \tau$ (τ - текущее время пребывания в реакторе), систему уравнений (4.7) можно представить в виде:

$$\frac{dc_i}{d\tau} = W_i \quad i = 1, 2..n \quad (4.8)$$

Если заменить правые части системы выражением (4.3), то **МО РИВ в стационарных условиях** в матричной форме примет вид

$$\left[\frac{dc_i}{d\tau} \right] = [a_{i,j}] \cdot [r_j] \quad i = 1, 2..n \quad (4.9)$$

Граничными условиями для решения системы (4.9) будут концентрации всех компонентов на входе в реактор (C_{exi}):

$$\tau = 0 \quad C_i = C_{exi} \quad i = 1, 2..n \quad (4.10)$$

Если принять, что в РИВ протекает реакция, рассмотренная в примере, то **МО РИВ**:

$$\begin{bmatrix} \frac{dC_1}{d\tau} \\ \frac{dC_2}{d\tau} \\ \frac{dC_3}{d\tau} \\ \frac{dC_4}{d\tau} \\ \frac{dC_5}{d\tau} \\ \frac{dC_6}{d\tau} \end{bmatrix} = \begin{bmatrix} -3 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 \\ 1 & -2 & 2 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & 1 & -1 & -1 & 1 \\ 0 & 0 & 0 & 2 & -2 \end{bmatrix} \cdot \begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \\ r_5 \end{bmatrix} \quad (4.11)$$

с граничными условиями (4.10).

Результатом решения системы уравнений **МО РИВ** являются профили концентраций по времени пребывания (по длине реактора).

Для разработки **МО реактора идеального смешения (РИС)** необходимо также составить материальный баланс по всем компонентам, участвующим в

химической реакции с учётом кинетических закономерностей (4.3), но на основе модели гидродинамики (4.6)

$$\frac{dc_i}{dt} = \frac{1}{\tau} \cdot (c_{\text{вх}i} - c_i) + W_i \quad i = 1, 2..n \quad (4.12)$$

Если заменить W_i в правой части системы (4.12) выражением (4.3), то **МО РИС в нестационарных условиях** в матричной форме примет вид

$$\left[\frac{dc_i}{dt} \right] = \frac{1}{\tau} \cdot [c_{\text{вх}i} - c_i] + [a_{i,j}] \cdot [r_j] \quad i = 1, 2..n \quad (4.13)$$

Начальными условиями для решения системы (4.13) будут концентрации всех компонентов в реакторе в начальный момент времени (C_{0i}):

$$t = 0 \quad C_i = C_{0i} \quad i = 1, 2..n \quad (4.14)$$

В частном случае, если моделируется процесс **пуска реактора**, $C_{0i} = 0$.

Если принять, что в РИС протекает реакция, рассмотренная в примере, то МО РИС:

$$\begin{bmatrix} \frac{dC_1}{dt} \\ \frac{dC_2}{dt} \\ \frac{dC_3}{d\tau} \\ \frac{dC_4}{dt} \\ \frac{dC_5}{dt} \\ \frac{dC_6}{dt} \end{bmatrix} = \frac{1}{\tau} * \begin{bmatrix} C_{\text{вх}1} - C_1 \\ C_{\text{вх}2} - C_2 \\ C_{\text{вх}3} - C_3 \\ C_{\text{вх}4} - C_4 \\ C_{\text{вх}5} - C_5 \\ C_{\text{вх}6} - C_6 \end{bmatrix} + \begin{bmatrix} -3 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 \\ 1 & -2 & 2 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & 1 & -1 & -1 & 1 \\ 0 & 0 & 0 & 2 & -2 \end{bmatrix} \cdot \begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \\ r_5 \end{bmatrix} \quad (15)$$

с начальными условиями (4.14).

Решением МО РИС является изменение концентраций компонентов в реакторе с течением времени. Моделирование процесса пуска реактора позволяет определить время выхода реактора на стационарный режим (время переходного процесса) и концентрации на выходе из реактора в стационарном режиме. В стационарном режиме концентрации всех реагентов постоянны, т.е. не зависят от времени.

Каскад реакторов представляет собой последовательно соединённые реакторы одного типа. Вход в каскад является входом в первый реактор, выход из последнего реактора является выходом из каскада. Выход из любого предыдущего (k-1)-го реактора является входом для последующего k-го реактора. **МО каскада химических реакторов идеального смешения (КРИС) в нестационарных условиях** представляет собой совокупность моделей РИС. Для любого k-го реактора КРИС МО имеет вид:

$$\left[\frac{dc_i^k}{dt} \right] = \frac{1}{\tau_k} \cdot [c_i^{k-1} - c_i^k] + [a_{i,j}] \cdot [r_j], \quad i = 1, 2, \dots, n \quad k = 1, 2, \dots, m \quad (4.16)$$

где k – текущий номер реактора, $\overline{\tau_k}$ – время пребывания в k-ом реакторе, m – число реакторов в каскаде,

$$\text{при } k=1 \quad c_i^{k-1} = c_{\text{вх}i}$$

Начальными условиями для решения системы (4.16) будут концентрации всех компонентов во всех реакторах каскада в начальный момент времени (C_{0i}^k):

$$t = 0 \quad C_i^k = C_{0i}^k \quad i = 1, 2, \dots, n \quad k = 1, 2, \dots, m \quad (4.17)$$

В частном случае, если моделируется процесс пуска реактора, $C_{0i}^k = 0$. Если принять, что в КРИС протекает реакция, рассмотренная в примере, то МО КРИС:

$$\begin{bmatrix} \frac{dC_1^k}{dt} \\ \frac{dC_2^k}{dt} \\ \frac{dC_3^k}{d\tau} \\ \frac{dC_4^k}{dt} \\ \frac{dC_5^k}{dt} \\ \frac{dC_6^k}{dt} \end{bmatrix} = \frac{1}{\tau_k} * \begin{bmatrix} C_1^{k-1} - C_1^k \\ C_2^{k-1} - C_2^k \\ C_3^{k-1} - C_3^k \\ C_4^{k-1} - C_4^k \\ C_5^{k-1} - C_5^k \\ C_6^{k-1} - C_6^k \end{bmatrix} + \begin{bmatrix} -3 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 \\ 1 & -2 & 2 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & 1 & -1 & -1 & 1 \\ 0 & 0 & 0 & 2 & -2 \end{bmatrix} \cdot \begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \\ r_5 \end{bmatrix} \quad (4.18)$$

с начальными условиями (4.17).

Решением МО КРИС является изменение концентраций компонентов в реакторах каскада с течением времени. Моделирование процесса пуска позволяет определить время выхода КРИС на стационарный режим (время переходного процесса) и концентрации в реакторах и на выходе из каскада в стационарном режиме.

2. РЕАЛИЗАЦИЯ МАТЕМАТИЧЕСКОГО ОПИСАНИЯ В SIMULINK

Решение поставленной задачи в Simulink осуществляется в соответствии с технологией, рассмотренной в предыдущих лабораторных работах. За основу может быть принята S-модель решения системы обыкновенных дифференциальных уравнений из лабораторной работы №1. Элементы вектора $[r_j]$ можно вычислить с помощью блоков Fcn (например, для решения систем (4.11), или (4.15), или (4.18) таких блоков будет 5), а объединить их в вектор - с помощью мультиплектора.

При составлении S-модели КРИС можно оформить S-модель РИС в виде подсистемы: вход подсистемы (блок In) устанавливается на информационном потоке входной концентрации, выход (блок Out) – на информационном потоке выходной концентрации (создание подсистем см. ниже в Примечании). Тогда S-модель КРИС можно представить как последовательное соединение подсистем. Очевидно, что число подсистем равно числу реакторов в каскаде. На вход первой подсистемы подаются входные концентрации, на выходе последней подсистемы снимаются концентрации на выходе из каскада.

Примечание. Для выделения части S-модели в подсистему необходимо с помощью мыши обрисовать общим контуром выделяемые блоки, затем в меню команды Edit выбрать Create subsystem.

После настройки параметров блоков и запуска модели на выполнение, следует получить результаты в графическом виде и в виде таблицы. Для определения времени завершения переходного процесса в РИС и КРИС следует увеличивать модельное время до достижения условия постоянства концентраций во времени. По результатам моделирования, **сравнивая концентрации продукта реакции на выходе из РИВ, РИС и КРИС в стационарных условиях (при одинаковом времени пребывания), можно выбрать эффективный реактор** для проведения рассматриваемой реакции.

Исходные данные для моделирования: механизм химической реакции, константы скорости стадий и концентрации на входе в реактор – в индивидуальном задании. Время пребывания в реакторах РИВ, РИС и КРИС принять равным 10 мин. Расчёты для КРИС выполнить для числа реакторов в каскаде $m = 3$ и $m = 5$.

3. ВАРИАНТЫ ЗАДАНИЙ

Таблица 4.1

Номер варианта	Механизм химических реакций
1	$2X_1 \rightarrow X_2, 2X_2 + X_4 \rightarrow X_5, X_1 \rightarrow X_3 + X_4, X_5 + X_3 \leftrightarrow 2X_6$
2	$X_1 \leftrightarrow 2X_2, X_2 \rightarrow X_3, X_3 + 2X_4 \rightarrow X_5, X_5 \rightarrow X_1 + X_6$
3	$3X_1 \rightarrow 2X_2 + X_3, 2X_3 + X_4 \leftrightarrow X_5, 2X_5 \leftrightarrow X_6$
4	$X_1 \rightarrow 3X_2, X_2 + X_3 + X_4 \rightarrow X_5, 2X_2 \leftrightarrow X_6$
5	$3X_1 \rightarrow 2X_2 + X_3, 2X_3 + X_4 \leftrightarrow X_5, 2X_5 \leftrightarrow 2X_6$
6	$X_1 \leftrightarrow 2X_2, X_2 \rightarrow X_3, X_3 + 2X_4 \rightarrow X_5, X_5 \rightarrow X_1 + X_6$
7	$2X_1 \rightarrow X_2, 2X_2 + X_4 \rightarrow X_5, X_1 \rightarrow X_3 + X_4, X_5 + X_3 \leftrightarrow 2X_6$
8	$3X_1 \rightarrow X_2, X_2 \rightarrow 2X_3 + X_4, X_4 + X_2 \rightarrow X_5, X_2 + X_1 \rightarrow X_6$
9	$X_1 + 2X_2 \rightarrow X_3, X_3 \rightarrow X_4 + X_5, X_5 \rightarrow X_6$
10	$X_1 + X_2 + X_3 \rightarrow X_4, X_4 + X_2 \rightarrow X_5, X_5 + X_3 \rightarrow X_6$
11	$2X_1 + X_2 \rightarrow X_3, 2X_3 \rightarrow X_4, X_4 + X_1 \rightarrow X_5, X_3 + X_5 \rightarrow X_6$
12	$X_1 + X_2 \rightarrow 2X_3, X_3 + X_4 \rightarrow 2X_5, X_5 \leftrightarrow 2X_6$
13	$X_1 \leftrightarrow 2X_2, X_2 + X_3 \rightarrow X_4, X_4 + X_3 \rightarrow X_5, X_5 + X_1 \rightarrow X_6$
14	$3X_1 \rightarrow X_2, X_2 \rightarrow 2X_3 + X_4, X_4 + X_2 \rightarrow X_5, X_2 + X_1 \rightarrow X_6$
15	$X_1 + 2X_2 \rightarrow X_3, X_3 \rightarrow X_4 + X_5, X_5 \rightarrow X_6$
16	$X_1 \leftrightarrow 2X_2, X_2 + X_3 \rightarrow X_4, X_4 + X_3 \rightarrow X_5, X_5 + X_1 \rightarrow X_6$

Таблица 4.2

Номер варианта	Концентрации на входе в реактор						Константы скорости стадий				
	C_{1_0}	C_{2_0}	C_{3_0}	C_{4_0}	C_{5_0}	C_{6_0}	K_1	K_2	K_3	K_4	K_5
1	2	0	0,2	0,5	0	0	0,5	0,8	0,2	0,45	0,1
2	0	0,5	0,8	1,6	0,1	0,1	0,45	0,2	0,25	0,4	0,3
3	1,5	0	0,2	1,2	0	0	0,1	0,36	0,25	0,3	0,5
4	1	1,6	0,8	1,2	0	0	0,35	0,5	0,7	0,3	
5	1,5	0	0,2	1,2	0	0	0,1	0,36	0,25	0,3	0,5
6	0	0,5	0,8	1,6	0,1	0,1	0,45	0,2	0,25	0,4	0,3
7	2	0	0,2	0,5	0	0	0,5	0,8	0,2	0,45	0,1
8	2	1	0	0,5	0,2	0,1	0,4	0,25	0,5	0,34	
9	0,8	2	0,5	0	0	0	0,1	0,5	0,35		
10	2	1	1,2	0,4	0,1	0	0,15	0,4	0,24		
11	1,8	0,9	0,5	0,2	0,5	0,4	0,15	0,4	0,3	0,25	
12	1	1	0	0,9	0	0,2	0,4	0,33	0,2	0,6	
13	1,6	0	1,2	0	0,3	0,5	0,1	0,5	0,35	0,4	0,2
14	2	1	0	0,5	0,2	0,1	0,4	0,25	0,5	0,34	
15	0,8	2	0,5	0	0	0	0,1	0,5	0,35		
16	1,6	0	1,2	0	0,3	0,5	0,1	0,5	0,35	0,4	0,2

4. ВОПРОСЫ ДЛЯ САМОКОНТРОЛЯ ЗНАНИЙ

1. Цель системного анализа (СА) гомогенного химического реактора.
2. Основные принципы СА.
3. Из каких подсистем состоит система «гомогенный химический реактор»?
4. Назовите основные признаки системы.
5. Элементы и связи в подсистеме «химическая реакция».
6. Каким образом разрабатывается математическое описание подсистемы «химическая реакция»?
7. Как составляется стехиометрическая матрица?
8. Элементы и связи в подсистеме «движение потока».
9. Какой подход используется для описания подсистемы «движение потока»?
10. Назовите типовые модели гидродинамики.
11. Перечислите допущения, принятые в типовых моделях гидродинамики.
12. Как осуществляется синтез математического описания гомогенного химического реактора?
13. Системой уравнений какого типа описывается нестационарный режим работы РИВ?
14. Системой уравнений какого типа описывается стационарный режим работы РИВ?
15. Что является результатом моделирования стационарного режима работы РИВ?
16. Системой уравнений какого типа описывается нестационарный режим работы РИС?
17. Что является результатом моделирования нестационарного режима работы РИС?
18. Системой уравнений какого типа описывается нестационарный режим работы каскада РИС?
19. Что является результатом моделирования нестационарного режима работы КРИС?
20. Какой ситуацией в работе реактора обусловлен нестационарный режим?
21. Начальные условия при моделировании РИВ, РИС и каскада РИС.
22. Каким образом разрабатывается s-модель КРИС?
23. Значение параметра какого блока изменяется в подсистемах s-модели КРИС при изменении числа реакторов в каскаде?

ЛАБОРАТОРНАЯ РАБОТА № 5.

ДИСКРЕТНО-ДЕТЕРМИНИРОВАННЫЕ И ДИСКРЕТНО-СТОХАСТИЧЕСКИЕ МОДЕЛИ. F И P-СХЕМЫ

ЦЕЛЬ РАБОТЫ: освоить технологию построения и исследования дискретно-детерминированных и дискретно-стохастических моделей с помощью *Simulink* и *Stateflow*.

1. ДИСКРЕТНО-ДЕТЕРМИНИРОВАННЫЕ МОДЕЛИ ИЛИ F-СХЕМЫ

Особенности дискретно-детерминированного подхода чаще всего рассматриваются на базе теории автоматов. Теория автоматов – раздел теоретической кибернетики, в котором изучаются математические модели – автоматы. Автомат перерабатывает дискретную информацию и меняет свои внутренние состояния в допустимые моменты времени. Если конечное множество объектов можно интерпретировать как алфавит, а сами объекты рассматривать как буквы этого алфавита, набор из этих букв представляет собой упорядоченную совокупность, называемую словом, то автомат представляет собой некий черный ящик, который преобразует буквы входного сигнала в буквы выходного сигнала. **Конечным автоматом** будем называть такой автомат, у которого множество состояний конечно. Для того чтобы задать конечный автомат (finite automata), необходимо задать одношаговую функцию перехода

$$z(t) = \varphi[z(t-1), x(t)]$$

и функцию выхода

$$y(t) = \psi[z(t-1), x(t)].$$

Автомат задается F-схемой вида:

$$F = \langle Z, X, Y, \varphi, \psi, Z_0 \rangle,$$

где Z – конечное множество состояний; X – входной сигнал (алфавит); Y – выходной алфавит; φ, ψ – функции перехода и выхода; Z_0 – начальное состояние алфавита

Автомат функционирует в дискретные моменты времени, которые называются тактами.

Любой автомат может быть задан с помощью таблиц переходов и выходов.

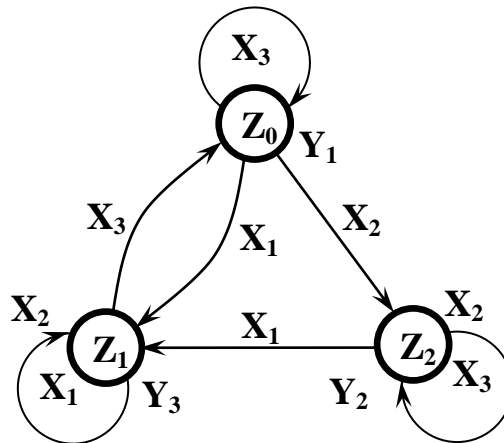
Таблица перехода:

X/Z	Z ₁	Z ₂	Z ₃	Z ₄
X ₁	Z ₂	Z ₃	Z ₁	Z ₂
X ₂	Z ₁	Z ₁	Z ₄	Z ₄
X ₃	Z ₂	Z ₃	Z ₂	Z ₁

Таблица выхода:

X/Z	Z ₁	Z ₂	Z ₃	Z ₄
X ₁	Y ₁	Y ₂	Y ₁	Y ₂
X ₂	Y ₁	Y ₁	Y ₂	Y ₁
X ₃	Y ₁	Y ₁	Y ₂	Y ₂

Автомат может быть задан с помощью ориентированного графа:



Различают несколько видов автоматов:

1. Автомат первого рода или автомат Мили

$$\begin{aligned}
 z(t+1) &= \varphi[z(t), x(t)] \\
 y(t) &= \psi[z(t), x(t)] \\
 t &= 0, 1, 2, \dots
 \end{aligned}$$

2. Автомат второго рода

$$\begin{aligned}
 z(t+1) &= \varphi[z(t), x(t)], \quad t = 0, 1, 2, \dots \\
 y(t) &= \psi[z(t), x(t-1)], \quad t = 1, 2, \dots
 \end{aligned}$$

Если при этом $y(t)=\psi[z(t)]$ есть выходная функция только состояния, то это автомат Мура.

Автоматы бывают с памятью и без памяти. Автоматы с памятью имеют более одного состояния, автоматы без памяти – только одно, и называются комбинационными или логическими. Автомат может быть синхронным: моменты поступления входных сигналов, изменения состояний и выдачи выходных сигналов заранее определены, и асинхронным: изменяет свои состояния при поступлении входных сигналов в произвольные моменты времени. Асинхронный автомат функционирует в непрерывном времени.

2. ДИСКРЕТНО-СТОХАСТИЧЕСКИЕ МОДЕЛИ ИЛИ Р-СХЕМЫ

Р-схемы основаны на понятии вероятностного или стохастического автомата. Р-автоматы являются более общим случаем конечных автоматов. Переход в определенные состояния и выдача выходного сигнала происходит на основании некоторой функции распределения вероятностей. Поэтому вероятностный автомат задается совокупностью матриц вероятностей вида:

$$P_{ij}(x) = P\{z(t) = z_j / z(t-1) = z_i; x(t) = x\},$$

где $P_{ij}(x)$ – условные вероятности того, что текущее состояние $z(t) = z_j$, при условии, что предыдущее состояние $z(t-1) = z_i$, а внешнее воздействие $x(t) = x$.

Условные вероятности составляют матрицу $\|P_{ij}(x)\| \rightarrow P_{i1}, P_{i2}, \dots, P_{in}$, причем

$\sum_{j=1}^n P_{ij} = 1$. Далее, в соответствие с вероятностями перехода в определенное состояние, по жребию выбирается конкретное состояние Z_j и вычисляется конкретный выходной сигнал.

Функция перехода $z(t) = \bar{\varphi}[z(t-1), x(t)]$, где $\bar{\varphi}$ – некоторая случайная функция. Для выбора состояния z_j из множества состояний Z по жребию может быть использован соответствующий случайный процесс (выбор пронумерованных фишек и т.п.). В ЭВМ этот процесс организуется в виде программного или аппаратурного датчика. Поскольку количество состояний конечно, а сумма вероятностей равна единице, то процедура всегда приводит к выбору некоторого конкретного состояния.

Автомат, у которого переход в определенное состояние определяется случайной величиной, называется автоматом со случайными переходами.

Если начальное состояние автомата определяется случайным образом, то он называется автоматом со случайными начальными состояниями, то есть состояние $z_0(t)$ выбирается по жребию с вероятностями $\pi = (\pi_1, \pi_2, \pi_3, \dots, \pi_n)$,

$$0 \leq \pi_i < 1 \quad \sum_{i=1}^n \pi_i = 1.$$

Если функция выходов определяет не конкретный выходной сигнал y , а лишь распределение вероятностей на множестве Y , то в этом случае мы имеем дело с автоматом со случайными выходами. Чтобы задать такой автомат, помимо функции перехода задается совокупность матриц $\|q_{ir}(x)\|$, элементы которых представляют условные вероятности того, что выходной сигнал $y(t) = y_r$ при условии, что $z(t-1) = z_i$, а выходной сигнал равен x

$$q_{ir}(x) = P\{y(t) = y_r / z(t-1) = z_i; x(t) = x\}.$$

3. ПРИМЕРЫ РЕШЕНИЯ ЗАДАЧ

Задача №1. Рассмотрим автомат для продажи билетов. Этот автомат принимает монеты достоинством 1, 2, 3 и 5 коп. и выдает билеты стоимостью 5 коп.

Решение. Данное устройство можно представить как конечный асинхронный автомат Мили с множеством состояний $Z = \{0, 1, 2, 3, 4\}$, входным алфавитом $X = \{1, 2, 3, 5\}$ и выходным алфавитом $Y = \{0, 1\}$, где 0 соответствует ситуации "билет не выдается", а 1 – ситуации "билет выдается". Функция переходов $\varphi(t)$ определяется соотношением

$$z(t) = (z(t-1) + x(t)) \bmod 5,$$

а функция выходов $\psi(t)$ – соотношением

$$y(t) = \begin{cases} 0, & \text{когда } z(t-1) + x(t) \leq 4 \\ 1, & \text{если } z(t-1) + x(t) > 4 \end{cases}$$

В качестве иллюстрации построим таблицы переходов и выходов рассматриваемого автомата:

Таблица переходов

x	Z				
	0	1	2	3	4
1	1	2	3	4	0
2	2	3	4	0	0
3	3	4	0	0	0
5	0	0	0	0	0

Таблица выходов

x	z				
	0	1	2	3	4
1	0	0	0	0	1
2	0	0	0	1	1
3	0	0	1	1	1
5	1	1	1	1	1

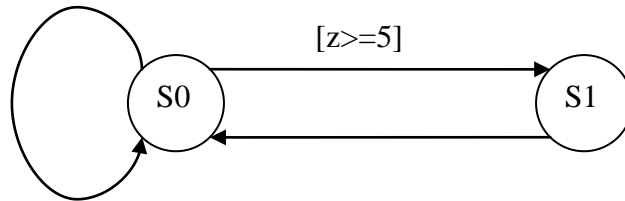
Задайте автомат в виде ориентированного графа самостоятельно.

Построение модели. Для реализации поставленной задачи удобно использовать расширение пакета **Simulink Stateflow**. Расширение **Stateflow** представляет собой концепцию теории автоматов посредством карт состояний Харела в собственной интерпретации. Подробную информацию о назначении, принципах использования и работы в **Stateflow** можно найти в справочной системе **MatLab**.

Вариант №1. Опишем автомат через два состояния: состояние S_0 - прием монет, и состояние S_1 - продажа билета. Таким образом, начальную постановку задачи в виде **автомата Мили** мы заменили **автоматом Мура**, у которого выходная функция зависит только от состояния. Входная переменная X может принимать значение 1, 2, 3 или 5. Выходная переменная Y – 0 или 1. Введем внутреннюю переменную Z – сумма денег в монетоприемнике. Переход из со-

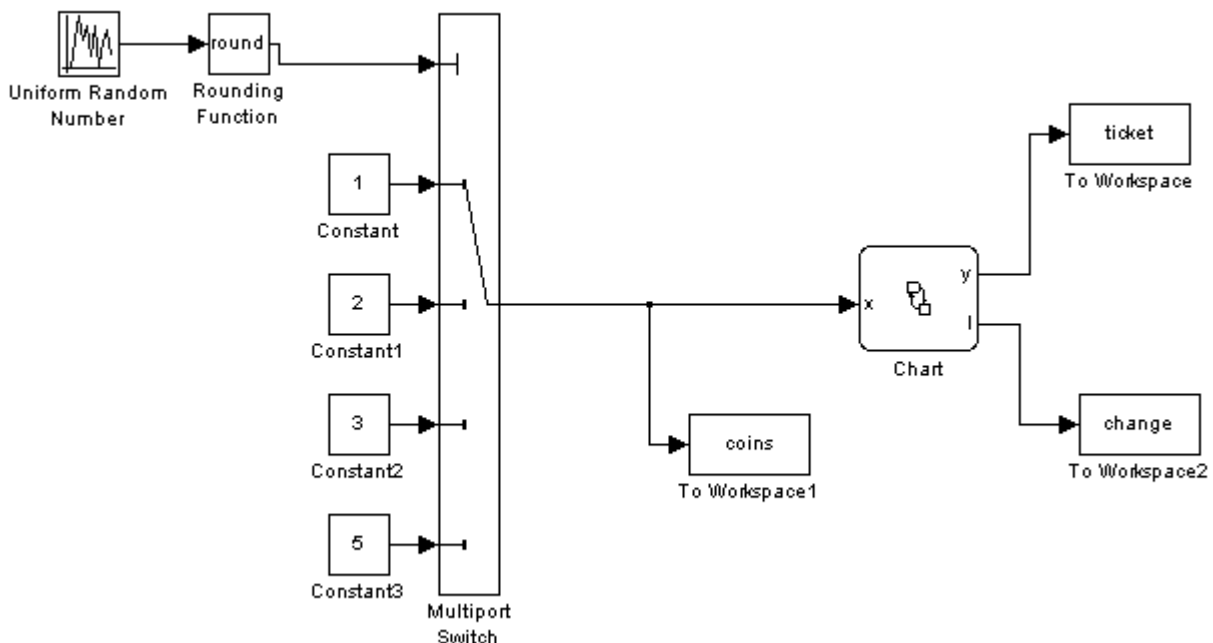
стояния S_0 в состояние S_1 происходит по условию ($Z \geq 5$). В начальный момент времени система находится в состоянии S_0 , а сумма денег $Z = 0$.

Задачу легко представить в виде графа:

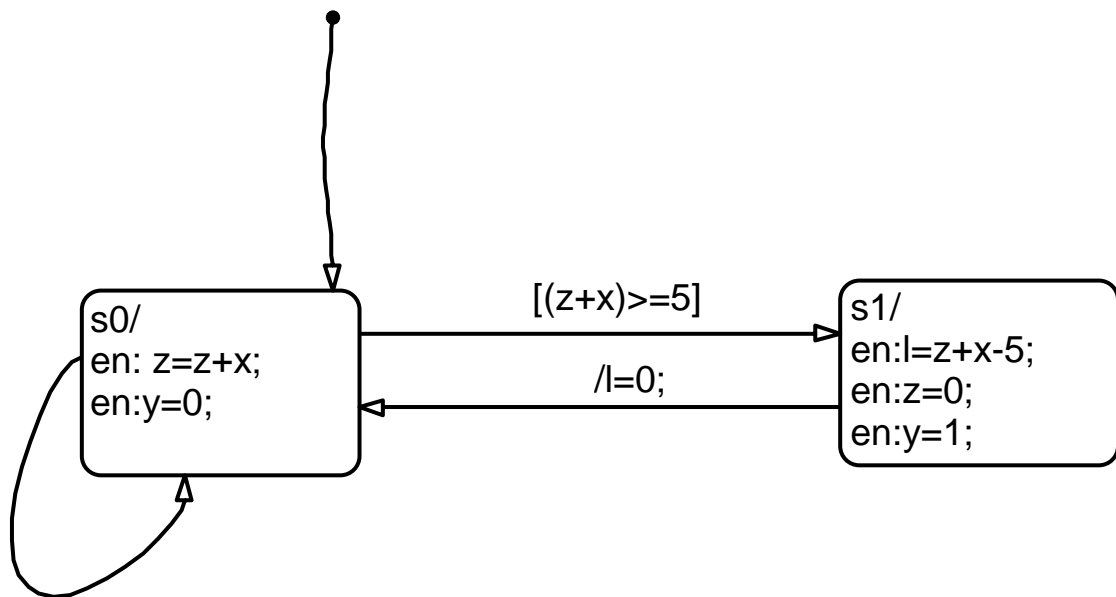


Построим модель в пакете **Simulink**. Соберите схему согласно рисунку.

Блок **Uniform Random Number** генерирует управляющее воздействие в диапазоне от 1 до 4. Блок **Rounding Function** округляет числа до целого. Блок **Multiport Switch** в зависимости от значения управляющего сигнала (1, 2, 3, 4) подключает соответствующий порт. В результате на вход блока **Chart** поступают сигналы в виде 1, 2, 3 или 5 коп. Блок **To Workspace1** сохраняет результат поступивших монет в переменной **coins**. Для данного блока, как и для всех блоков этого типа, необходимо установить параметр **Save format: Array**. Блок **To Workspace** выдает информацию о выдаче или не выдаче билета в данный такт времени, а блок **To Workspace2** – о сдаче, если таковая нужна. Блок **Chart** находится в библиотеке **Stateflow**. В первый момент его использования он пустой и не имеет никаких входов и выходов.



Двойным щелчком ЛКМ по блоку **Chart** войдите в режим его редактирования и сформируйте такую схему (см. рисунок).



en или **enter** – действие, совершаемое при входе в состояние.

В меню редактора выберите **Add – Data – Input from Simulink**. В появившемся окне в поле **Name** введите название входной переменной, в нашем случае **x**. Аналогично добавьте переменные **y** и **l**, т.е. **Add – Data – Output to Simulink**. И, наконец, определите внутреннюю переменную **z**, **Add – Data – Local**. Посмотреть все переменные можно в **Explore: Tools – Explore**. В принципе переменные можно определять непосредственно в **Explore**.

Вернитесь в S-модель и соедините появившиеся входы и выходы, согласно рисунку.

Установите параметры моделирования **Simulation – Simulation parameters**:

Start time – 1;
Stop time – 50;
Type – **Fixed step; discrete**;
Fixed step size – 1.

Теперь можно "проиграть" модель. Результаты моделирования сохраняются в **Workspace** в массивах с соответствующими названиями.

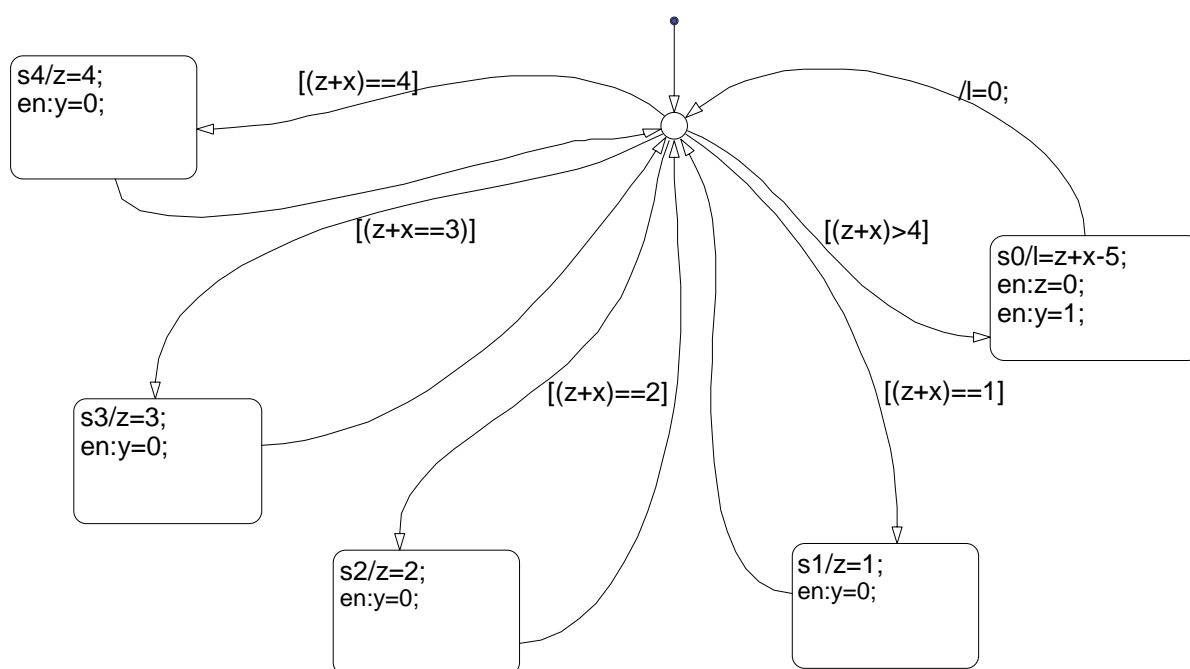
Для того чтобы использовать результаты в дальнейшем, можно предложить следующие способы их сохранения:

1. После выполнения моделирования все результаты сохраняются в рабочей области в соответствующих переменных. Наберите в командном окне следующую команду: **z = [tout, coins, ticket, change]**, а затем наберите имя созданной переменной и нажмите **Enter**. В командном окне появятся результаты. Теперь их легко скопировать в буфер обмена и сохранить в каком-нибудь текстовом документе.

2. Другой вариант в целом похож на первый, но результаты мы сразу сохраним в **ASCII** кодах в некотором внешнем файле, например **results.out**. Для

этого снова объединим все результаты в одну переменную **z**. Затем наберите команду: **diary results.out**. Нажмите **Enter**. Теперь все, что мы будем вводить в командном окне, будет записываться в указанный файл. Далее выведите значения переменной **z**, просто набрав в командном окне ее имя и нажав **Enter**. Теперь для завершения работы с файлом необходимо набрать команду: **diary off**. В результате связь с внешним файлом будет разорвана, а он станет доступен для работы с ним. Далее следует запустить **MS Excel** и открыть файл **results.out**. **MS Excel** запустит мастер конвертации. Следуя его указаниям, получим удобную для дальнейшей работы таблицу, содержащую в данном случае 4 столбца и 50 строк с результатами.

Вариант №2. Блок **Chart** можно реализовать и так (что соответствует исходной постановке задачи, т.е. автомату Мили):

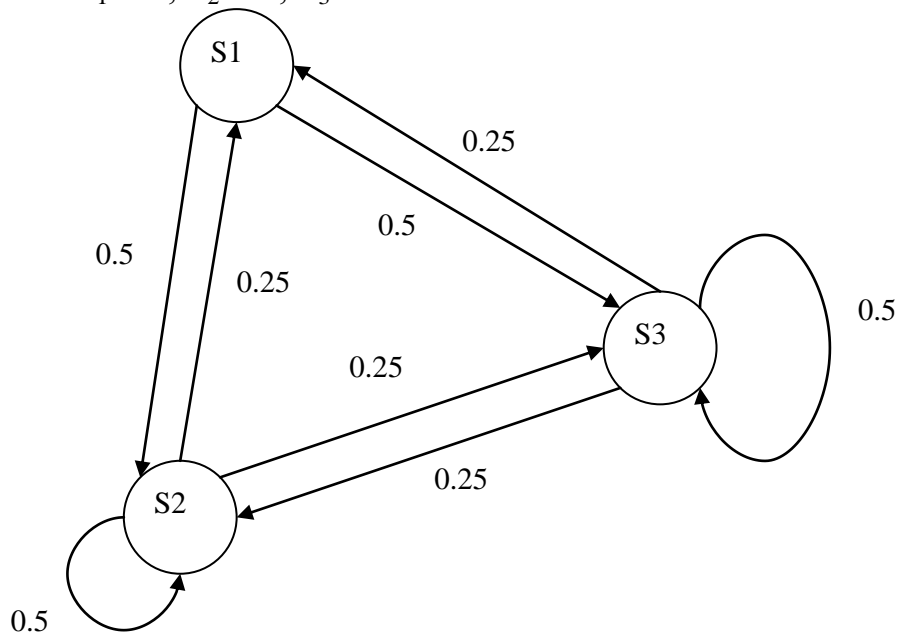


Задача №2. На Земле Оз погода бывает всего трех типов: дождь, солнечно или снег. Если сегодня солнечный день, то завтра будет дождь или снег с одинаковой вероятностью. Если сегодня - дождь (или снег), то половина шансов за то, что такая же погода будет завтра. Если же происходит изменение, то только в половине случаев оно приводит к солнечному дню. Определить вероятность появления всех типов погоды.

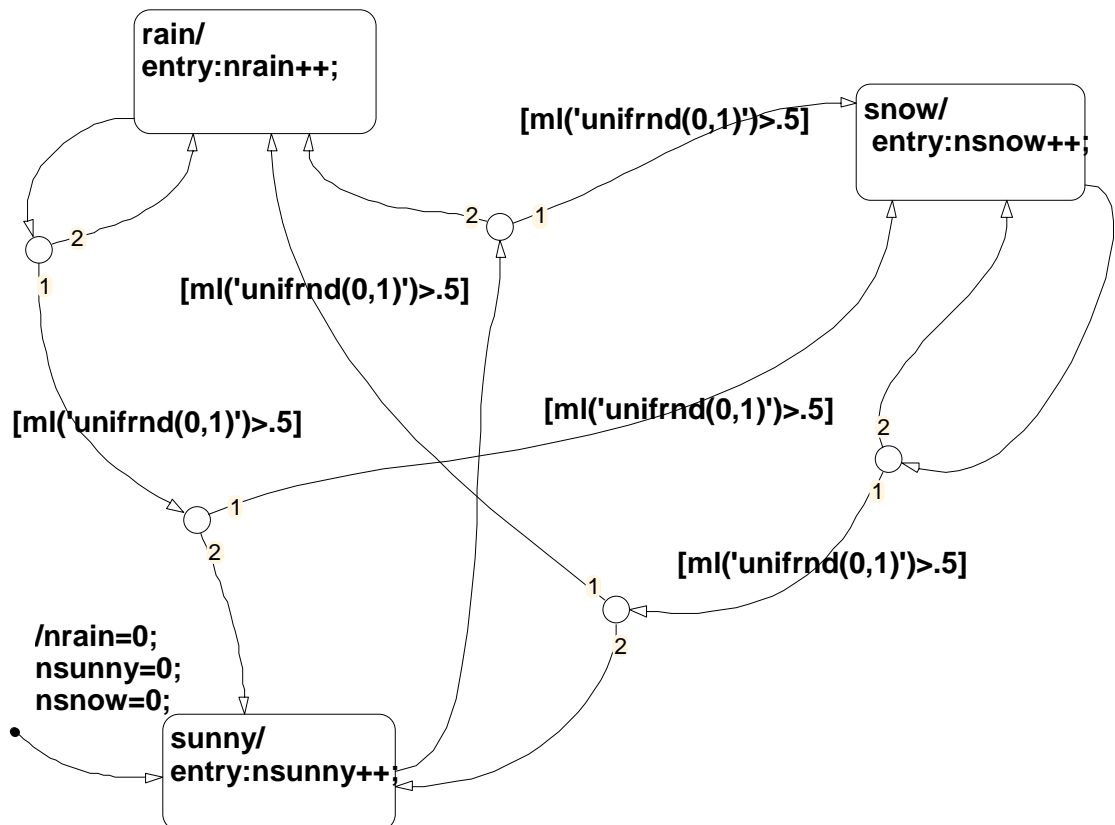
Решение. Данную задачу можно представить в виде вероятностного автомата со случайными переходами. Причем значения выходов совпадают со значениями состояний, т.е. имеем автомат Мура. Для того чтобы задать вероятностный автомат, надо определить матрицу переходных вероятностей $\|P_{ij}\|$. В нашем случае, как это следует из условия задачи, она будет иметь вид:

$$\|P_{ij}\| = \begin{vmatrix} 0 & 0.5 & 0.5 \\ 0.25 & 0.5 & 0.25 \\ 0.25 & 0.25 & 0.5 \end{vmatrix}$$

Образует эргодическую цепь Маркова с тремя состояниями **rain (S2)**, **sunny(S1)** и **snow(S3)** для дождя, солнца и снега соответственно. Начальные вероятности равны: $P_1 = 1$; $P_2 = 0$; $P_3 = 0$.



Данному графу будет соответствовать следующая Stateflow-диаграмма.

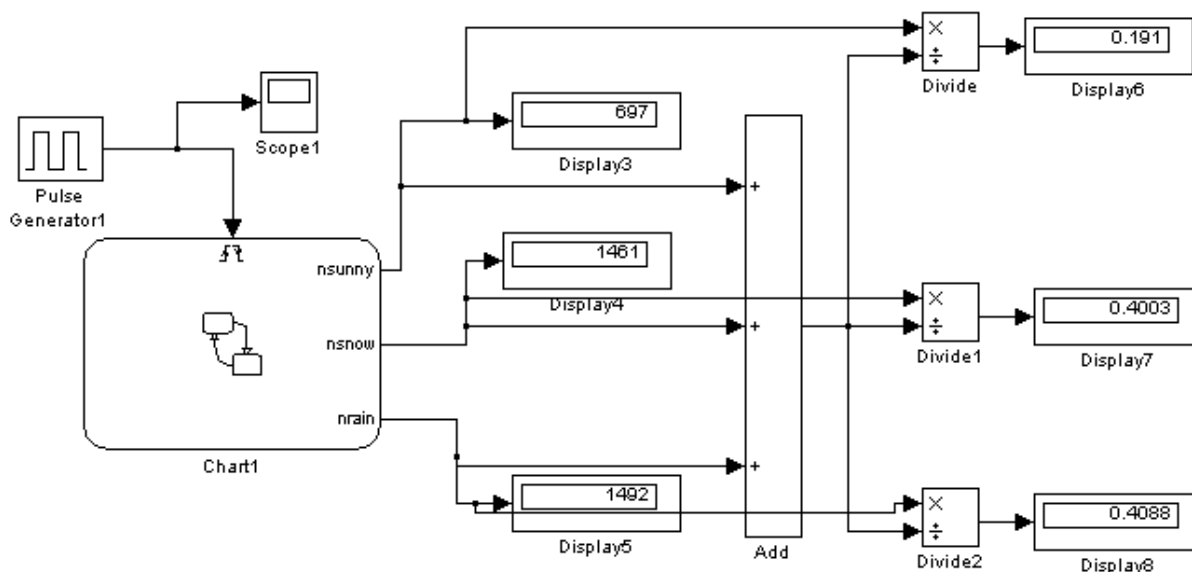


Начальное состояние - **sunny**, о чем свидетельствует наличие графического объекта переход по умолчанию (**Default transition**) к состоянию **sunny**. Этот переход сопровождается действием перехода (**Transition action**) **/nrain=0;nsunny=0;nsnow=0**. Это действие устанавливает в ноль счетчики количества дождливых, солнечных дней и дней, когда идет снег. Заметим, что в данном случае это действие является избыточным, так как начальные значения этих переменных равны нулю по умолчанию. В этом нетрудно убедиться, открыв проводник **Stateflow Explore** и просмотрев графу **InitVal**.

Name	Scope	Trigger	Type	Size	Min	Max	InitVal	FrWS	ToWS	Watch
event	Input(1)	Either					0			
nsunny	Output(1)		Real (double)				0			
nsnow	Output(2)		Real (double)				0			
nrain	Output(3)		Real (double)				0			

При входе в это состояние выполняется действие **nsunny++**, т.е. количество солнечных дней увеличивается на единицу.

Следующее событие **event** (смена суток организована в модели при помощи генератора прямоугольных импульсов **Pulse Generator**) переводит диаграмму в соединяемое подключение **Connective Junction**, откуда с вероятностью 0.5 диаграмма переходит в состояние **snow** и с вероятностью 0.5 - в состояние **rain**. Вероятностный переход основан на использовании условия **ml('unifrnd(0,1)')>.5** (вызов **MATLAB**-функции **unifrnd(0,1)**, т.е. генерация случайного числа из диапазона (0,1) и сравнение этого числа с числом 0.5). Остальные переходы организованы аналогичным образом в соответствии с логикой, описанной в задаче.



Результат работы модели на протяжении 10 лет модельного времени, как это следует из рисунка, дал 697 солнечных, 1461 снежных и 1492 дождливых дней. Аналитическое решение дает вероятность для солнечной погоды $1/5$, а для снега и дождя - $2/5$.

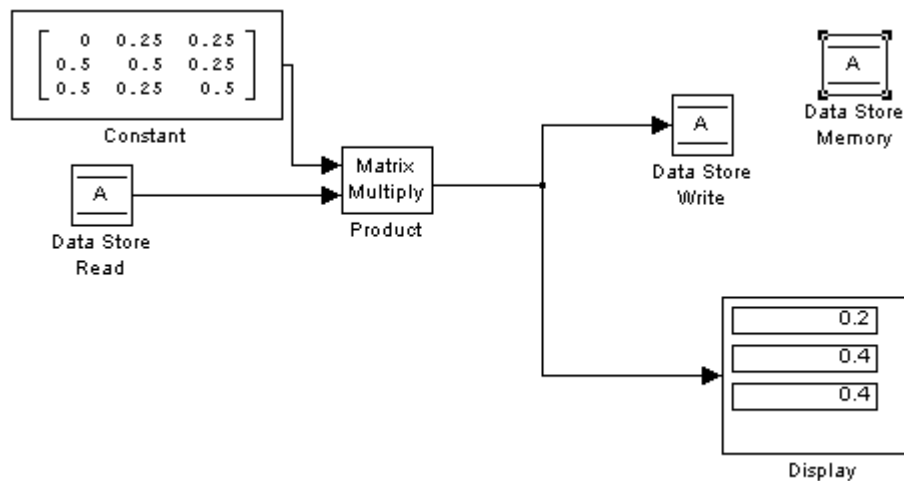
Аналитическое решение легко получить, используя положения теории Марковских цепей, согласно которой вероятность пребывания в i -ом состоянии на k -ом шаге определяется по формуле $P_i(k) = \sum_{j=1}^n P_j(k-1) \cdot P_{ji}$. То есть для нашего случая, получим систему уравнений вида:

$$\begin{aligned} P_1(k) &= P_2(k-1) \cdot P_{21} + P_3(k-1) \cdot P_{31} \\ P_2(k) &= P_1(k-1) \cdot P_{12} + P_2(k-1) \cdot P_{22} + P_3(k-1) \cdot P_{32} \\ P_3(k) &= P_1(k-1) \cdot P_{13} + P_2(k-1) \cdot P_{23} + P_3(k-1) \cdot P_{33} \end{aligned}$$

или в матричной форме

$$\begin{bmatrix} P_1(k) \\ P_2(k) \\ P_3(k) \end{bmatrix} = \begin{bmatrix} P_{11} & P_{21} & P_{31} \\ P_{12} & P_{22} & P_{32} \\ P_{13} & P_{23} & P_{33} \end{bmatrix} \times \begin{bmatrix} P_1(k-1) \\ P_2(k-1) \\ P_3(k-1) \end{bmatrix}$$

Решение данной системы удобнее проводить именно в матричной форме, для этого соберите S-модель, как показано на рисунке.



Обратите внимание на комплект новых блоков **Data Store Memory**, **Data Store Read**, **Data Store Write**. Эти блоки работают совместно, т.е. используют одну и ту же область памяти. В начальный момент моделирования в блок **Data Store Memory** необходимо поместить вектор начальных вероятностей $[1 \ 0 \ 0]$. На k -м шаге блок **Data Store Write** будет записывать текущие значения вероятностей P_i в блок **Data Store Memory**, откуда блок **Data Store Read** будет

их считывать для расчета вероятностей на $k+1$ шаге. Установите параметры моделирования **Simulation – Simulation parameters** такими:

Start time – 1;
Stop time – 100;
Type – Fixed step; discrete;
Fixed step size – 1.

4. ЗАДАЧИ ДЛЯ САМОСТОЯТЕЛЬНОГО ВЫПОЛНЕНИЯ

1. Выполнить задачу из примера №1 «Моделирование работы автомата по продаже билетов».
2. Выполнить приведенную ниже задачу согласно номеру варианта.

Задача 1.

В моменты времени t_1, t_2, t_3, \dots производится осмотр ЭВМ. Возможны следующие состояния ЭВМ: S_0 – полностью исправна; S_1 – незначительные неисправности, которые позволяют эксплуатировать ЭВМ; S_2 – существенные неисправности, дающие возможность решать ограниченное число задач; S_3 – ЭВМ полностью вышла из строя.

Матрица вероятностей перехода имеет вид

$$\|P_{ij}\| = \begin{vmatrix} 0.5 & 0.3 & 0.2 & 0 \\ 0 & 0.4 & 0.4 & 0.2 \\ 0 & 0 & 0.3 & 0.7 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

Постройте граф состояний. Найдите вероятности состояний ЭВМ после одного, двух, трех осмотров, если в начале ЭВМ была полностью исправна.

Задача 2.

Магазин продает две марки автомобилей А и В. Опыт эксплуатации этих марок свидетельствует о том, что для них имеют место различные матрицы переходных вероятностей, соответствующие состояниям «работает хорошо» (состояние 1) и «требует ремонта» (состояние 2):

$$\|P_A\| = \begin{vmatrix} 0.9 & 0.1 \\ 0.6 & 0.4 \end{vmatrix} \quad \text{и} \quad \|P_B\| = \begin{vmatrix} 0.8 & 0.2 \\ 0.7 & 0.3 \end{vmatrix}$$

Элементы матрицы перехода определены на годовой период эксплуатации.

Требуется:

1. Найти вероятности состояний для каждой марки автомобиля после 2х-летней эксплуатации, если в начальном состоянии автомобиль «работает хорошо».

2. Определить марку автомобиля, являющуюся более предпочтительной для покупки в личное пользование.

Задача 3.

Водитель такси обнаружил, что если он находится в городе А, то в среднем в 8 случаях из 10 он везет следующего пассажира в город Б, в остальных случаях будет поездка по городу А. Если же он находится в городе Б, то в среднем в 4 случаях из 10 он везет следующего пассажира в город А, в остальных же случаях будет поездка по городу Б.

Требуется:

1. Перечислить возможные состояния процесса и построить граф состояний.
2. Записать матрицу переходных состояний.
3. Найти вероятности состояний после двух шагов процесса, если:
 - а. В начальном состоянии водитель находился в городе А.
 - б. В начальном состоянии водитель находился в городе Б.
4. Найти вероятности состояний в установившемся режиме.

Задача 4.

Для многих экономических задач (энергетики, мелиорации и т.д.) необходимо знать чередование годов с определенными значениями годовых стоков рек. Конечно, это чередование не может быть определено абсолютно точно. Для определения вероятностей чередования (перехода) разделим стоки (элементы процесса), введя четыре градации (состояния): первую, вторую, третью и четвертую. В результате накопления влаги (в земле, водохранилищах и т. д.) будем для определенности считать, что за первой градацией (самый низкий сток) никогда не следует четвертая (самый высокий сток), а за четвертой — первая. Допустим, что остальные переходы возможны и:

из первой градации можно попасть в каждую из средних градаций вдвое чаще, чем опять в первую.

из четвертой градации переходы во вторую и третью градации бывают в четыре и пять раз чаще, чем возвращение в четвертую градацию;

из второй градации переход в другие градации может быть только реже: в первую — в два раза, в третью — на 25 %, в четвертую — в четыре раза, чем переход во вторую;

из третьей градации переход во вторую градацию столь же вероятен, как и возвращение в третью градацию, а переходы в первую и четвертую градации бывают в четыре раза реже.

Требуется:

1. Построить граф состояний.
2. Записать матрицу переходных состояний.
3. Найти вероятности чередования градаций в установившемся режиме.

Задача 5.

Матрица вероятностей перехода цепи Маркова имеет вид

$$\|P_{ij}\| = \begin{vmatrix} 0.1 & 0.5 & 0.4 \\ 0.6 & 0.2 & 0.2 \\ 0.3 & 0.4 & 0.3 \end{vmatrix}$$

Распределение по состояниям в момент $t = 0$ определяется вектором $q = (0.7; 0.2; 0.1)$.

Найти:

1. Распределение по состояниям в момент $t = 2$;
2. Вероятность того, что в моменты $t = 0, 1, 2, 3$ состояниями цепи будут соответственно 1, 3, 3, 2;
3. Стационарное распределение.

Задача 6.

Техническое устройство состоит из двух узлов и может находиться в одном из следующих состояний: 1) оба узла исправны; 2) неисправен первый узел; 3) неисправен второй узел и 4) оба узла неисправны. Вероятность выхода из строя после месячной эксплуатации для 1-го узла – $P_1 = 0.4$; для 2-го – $P_2 = 0.3$; обоих $P_{1,2} = 0.1$. В исходном состоянии оба узла исправны. Найдите переходные вероятности и вероятности состояний после двухмесячной эксплуатации, считая, что вероятность того, что узел будет отремонтирован, равна 1.

Задача 7.

В моменты времени t_1, t_2, t_3, \dots производится ревизия железнодорожного полотна на определённом участке. Возможны следующие состояния ж/д полотна: S_0 – полотно отвечает всем требованиям; S_1 – имеются незначительные нарушения, которые позволяют эксплуатировать полотно; S_2 – обнаружены неисправности, при которых участок должен быть закрыт.

Матрица вероятностей перехода имеет вид

$$\|P_{ij}\| = \begin{vmatrix} 0.8 & 0.1 & 0.1 \\ 0 & 0.6 & 0.4 \\ 0 & 0 & 1 \end{vmatrix}$$

Постройте граф состояний. Найдите вероятности состояний ж/д полотна после одного, двух, трех осмотров, если в начале полотно отвечало всем требованиям.

Задача 8.

Имеются три конкурирующих изделия X_0 , X_1 , X_2 . С целью определения спроса на эти изделия произведён опрос 100 человек. Оказалось, что изделие X_0 покупает 50 человек, изделие X_1 – 20 человек, а X_2 – 30 человек.

По истечении месяца оказалось, что из 50 покупателей изделия X_0 45 человек продолжают его покупать, 4 человека стали покупать изделие X_1 и 1 – изделие X_2 .

Из 20 покупателей изделия X_1 6 человек продолжают его покупать, 8 стали покупать изделие X_0 , 6 – изделие X_2 .

Из 30 покупателей изделия X_2 6 человек продолжают его покупать, 21 человек стал покупать изделие X_0 , 3 – изделие X_1 .

Найдите переходные вероятности и определите, какое изделие по истечении месяца будет пользоваться наибольшим спросом.

Задача 9.

Имеются три Интернет-провайдера (ИП1, ИП2, ИП3). С целью определения популярности провайдеров произведён опрос 300 пользователей Интернета. Оказалось, что 80 человек пользуются услугами ИП1, 120 – услугами ИП2 и 100 человек – услугами ИП3. По истечении месяца оказалось, что из 80 пользователей ИП1 60 человек не поменяли провайдера, 10 человек перешло к провайдеру ИП2 и 10 – к провайдеру ИП3.

Из 120 клиентов у ИП2 осталось 80, 30 перешли к ИП1 и 10 – к ИП3.

Из 100 клиентов у ИП3 осталось 50, 20 перешли к ИП1 и 30 – к ИП2.

Требуется определить, какой Интернет-провайдер будет иметь наибольшее число клиентов через месяц, через 6 месяцев.

Задача 10.

Фирма Проект-2000 имеет три продовольственных магазина: Экстра, Экстра-универмаг и Экстра-эконом. С целью определения предпочтений покупателей было опрошено 200 человек. Выяснилось, что 50 человек пользуются наиболее часто магазином Экстра, 60 – магазином Экстра-универмаг и 90 – отдают предпочтение Экстра-эконом. По истечении квартала этим же людям предложили вновь ответить на тот же вопрос. Оказалось, что из 50 покупателей Экстры её клиентами остались 30 человек, 15 ушли в Экстра-эконом и 5 – в Экстра-универмаг. Из 60 покупателей Экстра-универмаг 1 перешёл в магазин Экстра и 20 – в Экстра-эконом.

Определить, какому из магазинов покупатели отдадут предпочтение через квартал, через год.

Задача 11.

Тестируются сетевые операционные системы двух фирм-разработчиков. Рассматриваются два состояния системы: ОС работает и требуется перезагрузка ОС. По результатам тестирования в течение месяца получены матрицы переходных вероятностей для ОС1 и ОС2:

$$\|P_{oc1}\| = \begin{vmatrix} 0.85 & 0.15 \\ 0.4 & 0.6 \end{vmatrix} \quad \text{и} \quad \|P_{oc2}\| = \begin{vmatrix} 0.9 & 0.1 \\ 0.6 & 0.4 \end{vmatrix}$$

Элементы матрицы перехода определены на период эксплуатации в течение месяца.

Требуется:

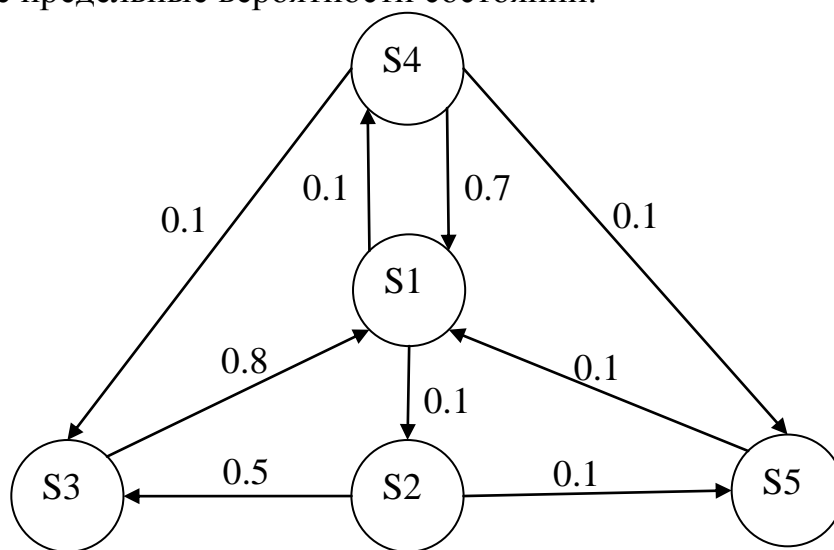
1. Найти вероятности состояний для каждой ОС после полугодовой эксплуатации, если в начальном состоянии ОС работает.
2. Определить наиболее надёжную ОС.

Задача 12.

Вычислительная машина находится в одном из следующих состояний:

- S1 – исправно работает;
- S2 – неисправна, тестируется;
- S3 – неисправна, настраивается программное обеспечение;
- S4 – находится на профилактике;
- S5 – ремонтируется, модернизируется.

Размеченный граф состояний показан на рис. Составьте систему уравнений и найдите предельные вероятности состояний.



Задача 13.

Организация по прокату автомобилей в городе выдает автомобили напрокат в трех пунктах города: А, В, С. Клиенты могут возвращать автомобили в любой из трех пунктов. Анализ процесса возвращения автомобилей из проката в течение года показал, что клиенты возвращают автомобили в пункты проката в соответствии со следующими вероятностями:

Пункты выдачи	Пункты приема		
	А	В	С
А	0,8	0,2	0

В	0,2	0	0,8
С	0,2	0,2	0,6

Требуется:

1. В предположении, что число клиентов в городе не изменяется, найти процентное распределение клиентов, возвращающих автомобили по станциям проката к концу года, если в начале года оно было равномерным.
2. Найти вероятности состояний в установившемся режиме.
3. Определить пункт проката, у которого более целесообразно строить станцию по ремонту автомобилей.

Задача 14.

В городе издаются три журнала: С1, С2, С3, и читатели выписывают только один из них. Пусть в среднем читатели стремятся поменять журнал, т.е. подписаться на другой не более одного раза в год, и вероятности таких изменений постоянны. Результаты маркетинговых исследований спроса читателей на журналы дали следующее процентное соотношение:

80% читателей С1 подписываются на С2;

15% читателей С2 подписываются на С3;

8% читателей С3 подписываются на С1.

Требуется:

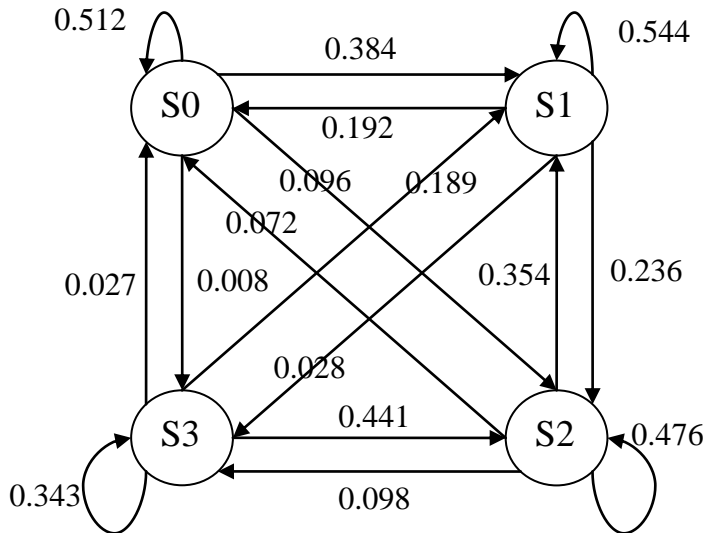
- Записать матрицу переходных вероятностей для средних годовых изменений.
- Предположить, что общее число подписчиков в городе постоянно, и определить, какая доля из их числа будет подписываться на указанные журналы через два года, если по состоянию на 1 января текущего года каждый журнал имел одинаковое число подписчиков.
- Найти вероятности состояний в установившемся режиме и определить журнал, который будет пользоваться наибольшим спросом у читателей.

Задача 15.

Возможны три градации урожайности: низкая, средняя и высокая. Если год имел среднюю урожайность, то с вероятностью 0.6 следующий год будет таким же, с вероятностью 0.3 – малоурожайным, либо следующий год будет таким же, как и предыдущий. Если год был высокоурожайным, то с равной вероятностью следующий будет либо неурожайным, либо со средней урожайностью. Если же год был неурожайным, то 80 % за то, что год будет со средней урожайностью, 15 % - с высокой урожайностью и 5 % за то, что будет неурожай. Определить сколько лет в течение столетия будет неурожай, средний урожай и высокий урожай.

Задача 16.

В локальной вычислительной сети работают три ЭВМ. Через равные промежутки времени (через месяц) все ЭВМ тестируются, в результате чего каждая признается либо исправной, либо требующей ремонта. Используя данные на рисунке, определите распределение вероятностей через год.



5. ВОПРОСЫ ДЛЯ САМОКОНТРОЛЯ ЗНАНИЙ

1. Что означает понятие дискретно-детерминированные модели?
2. Что означает понятие дискретно-стохастические модели?
3. Что такое конечный автомат?
4. Каким образом можно задать конечный автомат?
5. Какие способы задания конечного автомата существуют? В чем их преимущества и недостатки?
6. Чем отличается конечный автомат Мили от конечного автомата Мура?
7. Что такое Р-автоматы?
8. Как задаётся Р-автомат?
9. Какие виды вероятностных автоматов существуют?
10. Какова область применения моделей на основе конечных автоматов?
11. Какие инструменты в Matlab реализуют концепцию конечных автоматов?
12. Какие графические элементы используются для создания моделей конечных автоматов?
13. Что такое состояние автомата и каковы его свойства?
14. Какие виды действий и деятельности допустимы в состоянии?
15. Что такое переход? Какова его семантика? Какова сигнатура перехода?
16. Какую функцию выполняют соединения?

ЛАБОРАТОРНАЯ РАБОТА № 6.

МОДЕЛИРОВАНИЕ СЛУЧАЙНЫХ ВЕЛИЧИН, СОБЫТИЙ И ПРОЦЕССОВ

ЦЕЛЬ РАБОТЫ: освоение технологии моделирования случайных величин, событий и процессов.

1. МЕТОД СТАТИСТИЧЕСКИХ ИСПЫТАНИЙ

В тех случаях, когда при моделировании необходимо учитывать некоторый случайный фактор (элемент или явление), который невозможно описать аналитически, используют метод моделирования, называемый *методом статистических испытаний* или *методом Монте-Карло*. С помощью этого метода может быть решена любая вероятностная задача. Однако использовать его целесообразно в том случае, если решить задачу этим методом проще, чем любым другим.

Суть метода состоит в том, что вместо описания случайных явлений аналитическими зависимостями проводится розыгрыш случайного явления с помощью некоторой процедуры, которая дает случайный результат. С помощью розыгрыша получают одну реализацию случайного явления. Осуществляя многократно такой розыгрыш, накапливают статистический материал (то есть множество реализаций случайной величины), который можно обрабатывать статистическими методами.

Методом статистических испытаний можно оценить математическое ожидание и другие вероятностные характеристики. Например, оценку математического ожидания площади поражения цели можно определить как

$$M(S) = \frac{1}{N} \sum_{i=1}^N S_i. \quad (6.1)$$

При $N \rightarrow \infty$ оценка будет приближаться к математическому ожиданию в соответствии с законом больших чисел.

Алгоритм метода статистических испытаний такой:

1. Определить, что собой будет представлять испытание или розыгрыш.
2. Определить, какое испытание является успешным, а какое - нет.
3. Провести большое количество испытаний.
4. Обработать полученные результаты статистическими методами и рассчитать статистические оценки искомых величин.

К недостаткам метода можно отнести необходимость проведения большого количества испытаний, чтобы получить результат с заданной точностью.

Таким образом, метод статистических испытаний - это метод математического моделирования случайных величин, в котором сама случайность непосредственно включена в процесс моделирования и является его важным эле-

ментом. Каждый раз, когда в ход выполнения некоторой операции вмешивается случайный фактор, его влияние моделируется с помощью розыгрыша.

Для эффективного розыгрыша случайных величин используют генераторы случайных чисел. Такие генераторы строятся аппаратными и программными методами. Наиболее применимыми являются программные методы, которые дают возможность получить последовательности псевдослучайных чисел по рекуррентным формулам. Обычно используется мультипликативный конгруэнтный метод, рекуррентное соотношение для которого имеет вид:

$$X_{i+1} = aX_i \pmod{m}, \quad (6.2)$$

где a и m - некоторые константы. Необходимо взять последнее псевдослучайное число X_i , умножить его на постоянный коэффициент a и взять модуль полученного числа по m , то есть разделить на m и получить остаток. Этот остаток и будет следующим псевдослучайным числом X_{i+1} . Для двоичного компьютера $m = 2^g - 1$, где g - длина разрядной сетки. Например, для 32-разрядного компьютера $m = 2^{31} - 1 = 2147483647$, поскольку один разряд задает знак числа.

2. МОДЕЛИРОВАНИЕ ДИСКРЕТНЫХ СЛУЧАЙНЫХ ВЕЛИЧИН

Моделирование события. Пусть необходимо смоделировать, появление некоторого события A , вероятность наступления которого равняется $P(A) = P$. Обозначим обращения к генератору, который разыгрывает псевдослучайные, равномерно распределенные на интервале $(0, 1)$ числа r_i , через R . Событие A при розыгрыше будет наступать тогда, когда $r \leq P$, в противном случае происходит событие \bar{A} с вероятностью $r > P$.

Действительно:

$$P(r < P) = \int_0^P f(x) dx = \int_0^P f(r) dr = P = P(A). \quad (6.3)$$

Моделирование группы несовместных событий. Пусть есть группа несовместимых событий A_1, A_2, \dots, A_k . Известны вероятности вступления событий $P(A_1), P(A_2), \dots, P(A_k)$. Тогда из-за несовместности событий $\sum_{i=1}^k P(A_i) = 1$.

Пусть $p_i = P(A_i)$, $p_0 = 0$.

Если полученное число попало в интервал от $\sum_{k=0}^{i-1} p_k$ до $\sum_{k=0}^i p_k$, то произошло событие A_i . Такую процедуру называют определением результата испытания по жребию, и она основывается на формуле

$$P\left\{\sum_{k=0}^{i-1} p_k < r \leq \sum_{k=0}^i p_k\right\} = p_i = P(A_i), \quad (6.4)$$

где $p_0 = 0$.

Моделирование случайной дискретной величины. Моделирование случайной дискретной величины выполняется аналогично моделированию группы несовместимых событий. Дискретная случайная величина задается в соответствии с табл. 1.

Случайную величину X можно представить как полную группу событий:
 $A_1 = (X = x_1), A_2 = (X = x_2), \dots, A_n = (X = x_n)$

Возможное значение	x_1	x_2	...	x_n
Вероятность	p_1	p_2	...	p_n

Моделирование условного события. Моделирование условного события A , которое происходит при условии, что наступило событие B с вероятностью $P(A/B)$, выполняется следующим образом. Сначала моделируем событие B . Если событие B происходит, то моделируем наступление события A , если имеем \bar{B} , то не моделируем наступление события A .

3. МОДЕЛИРОВАНИЕ НЕПРЕРЫВНЫХ СЛУЧАЙНЫХ ВЕЛИЧИН

В данном случае используется метод обратной функции. Пусть есть некоторая функция распределения случайной величины. Разыграем на оси ординат точку r , используя функцию $F(x)$. Тогда можем получить значение величины x такое, что $F(x)=r$.

Найдем функцию распределения $F(x)$ случайной величины X . По определению она равна вероятности $P(X < x)$. Очевидно, что

$$P(X < x) = P(R < F(X)) = \int_0^{F(x)} f(r) dr = F(x). \quad (6.5)$$

Таким образом, последовательность r_1, r_2, r_3, \dots , принадлежащая $R(0, 1)$, преобразуется в последовательность x_1, x_2, x_3, \dots , которая имеет заданную функцию плотности распределения $f(x)$.

Моделирование равномерного распределения в интервале (а, b) случайной величины. Для моделирования воспользуемся методом обратной функции. Находим функцию распределения и приравниваем ее к случайному числу

$$R = \int_a^x \frac{dx}{b-a} = \frac{x-a}{b-a}. \quad (6.6)$$

Отсюда $x = (b - a)R + a$.

Моделирование экспоненциального распределения случайной величины. Функция плотности экспоненциального распределения случайной величины есть $f(x) = \lambda e^{-\lambda x}$. Воспользуемся методом обратной функции:

$$R = \int_0^x f(x) dx = \int_0^x \lambda e^{-\lambda x} dx = 1 - e^{-\lambda x}. \quad (6.7)$$

Из данного выражения находим x :

$$x = -\frac{1}{\lambda} \ln(1 - R). \quad (6.8)$$

Можно показать, что случайная величина $(1-R)$ распределена так же, как и величина R . Тогда, сделав замену $(1 - R)$ на R , получаем

$$x = -\frac{1}{\lambda} \ln R. \quad (6.9)$$

Моделирование нормального закона распределения случайной величины. Для моделирования нормального закона распределения случайной величины нельзя непосредственно воспользоваться методом обратной функции, поэтому используем центральную предельную теорему. Пусть случайная величина имеет математическое ожидание m_x и среднеквадратичное отклонение σ_x , а случайная величина Z имеет математическое ожидание $m_z = 0$ и среднеквадратичное отклонение $\sigma_z = 1$. Легко показать, что

$$X = \sigma_x Z + m_x. \quad (6.10)$$

Сформулируем центральную предельную теорему.

Если X_1, \dots, X_n - независимые случайные величины со средним значением $E[X_i] = a$, $i = \overline{1, n}$ и дисперсией $D[X_i] = \sigma^2$, $i = \overline{1, n}$, то при неограниченном увеличении n функция распределения случайной величины

$$\bar{X}^*(n) = \frac{\frac{1}{n}(X_1 + \dots + X_n) - a}{\frac{\sigma}{\sqrt{n}}} = \frac{(\bar{X}(n) - a)\sqrt{n}}{\sigma} \quad (6.11)$$

приближается к функции распределения стандартного нормального закона $\Phi(z)$ при всех значениях аргумента, то есть

$$F_{\bar{X}^*(n)} \rightarrow \Phi(z), \quad (6.12)$$

где $\Phi(z) = \frac{1}{\sqrt{2\pi}} \int_0^z e^{-\frac{z^2}{2}} dz$.

Для получения нормального закона распределения случайной величины достаточно суммировать шесть случайных величин, полученных с помощью генератора случайных чисел R , и, пронормировав полученные значения так, чтобы определить Z , по формуле найти значение X .

Обычно суммируют 12 случайных величин R_i , $Z = \sum_{i=1}^{12} R_i$, тогда дисперсия $D(Z)$ будет равняться единице.

4. СБОР СТАТИСТИЧЕСКИХ ДАННЫХ ДЛЯ ПОЛУЧЕНИЯ ОЦЕНОК ХАРАКТЕРИСТИК СЛУЧАЙНЫХ ВЕЛИЧИН

Основными элементами, из совокупности которых складывается вероятностная модель метода статистических испытаний, являются случайные реализации. Очевидно, что при решении некоторой задачи определения характеристик или параметров исходного случайного процесса должен быть определен этот случайный процесс.

Искомые величинами при использовании метода статистических испытаний являются оценки:

- вероятности наступления некоторого события;
- математического ожидания случайной величины;
- дисперсии случайной величины;
- коэффициентов ковариации или корреляции случайной величины.

Для оценки вероятности p наступления некоторого события A используется частота наступления этого события

$$p' = \frac{m}{N}, \tag{6.13}$$

где m - частота наступления события, а N - число опытов.

Для оценки математического ожидания случайной величины используется среднее значение

$$x' = E'[x] = \frac{1}{N} \sum_{i=1}^N x_i, \tag{6.14}$$

где x_i - i -я реализация случайной величины.

Для оценки дисперсии случайной величины ξ , используют формулу

$$S^2 = \frac{1}{N-1} \sum_{k=1}^N (x_k - \bar{x})^2, \quad (6.15)$$

где S^2 - оценка дисперсии случайной величины ξ .

Непосредственно использовать эти формулы для вычисления дисперсии сложно, поскольку среднее значение изменяется по мере накопления x_i , то есть нужно запоминать все N значений x_i . Поэтому для вычисления используют формулу:

$$S^2 = \overline{\sigma^2} = \frac{1}{N-1} \left[\sum_{i=1}^N x_i^2 - \frac{1}{N} \left(\sum_{i=1}^N x_i \right)^2 \right]. \quad (6.16)$$

В этом случае достаточно накапливать две суммы значений – x_i и x_i^2 .

Непосредственное использование этой формулы для программирования может привести к переполнению разрядной сетки, если запрограммировать ее в таком порядке, в котором она записана. Необходимо изменить последовательность действий, чтобы избавиться от очень больших чисел и переполнения разрядной сетки компьютера.

Все статистические оценки должны иметь определенные качественные показатели, к которым относятся *несмещенность*, *эффективность* и *состоятельность оценки*.

Для случайных величин ξ и η с возможными значениями x_k , y_k оценка корреляционного момента определяется так

$$\overline{K}_{\xi\eta} = \frac{1}{N-1} \left[\sum_{k=1}^N (x_k - \bar{x})(y_k - \bar{y}) \right] \quad (6.17)$$

или в удобной для вычислений форме

$$\overline{K}_{\xi\eta} = \frac{1}{N-1} \left[\sum_{k=1}^N x_k y_k - \frac{1}{N} \sum_{k=1}^N x_k \sum_{k=1}^N y_k \right]. \quad (6.18)$$

5. ОПРЕДЕЛЕНИЕ КОЛИЧЕСТВА РЕАЛИЗАЦИЙ ПРИ МОДЕЛИРОВАНИИ СЛУЧАЙНЫХ ВЕЛИЧИН

Число испытаний N определяет точность получаемых результатов моделирования. Если необходимо оценить величину параметра \mathbf{a} по результатам моделирования x_i , то за оценку следует брать величину \bar{x} , которая выступает в функции от x_i .

Из-за случайности \bar{x} будет отличаться от \mathbf{a} , то есть

$$|a - \bar{x}| < \varepsilon, \quad (6.19)$$

где ε - точность оценки. Вероятность того, что данное неравенство выполняется, обозначим через α :

$$P(|a - \bar{x}| < \varepsilon) = \alpha. \quad (6.20)$$

Данное выражение необходимо использовать для определения точности результатов статистических испытаний.

Определение количества реализаций для оценки вероятности наступления события. Пусть целью моделирования будет определение вероятности наступления некоторого события A , определяющего состояние моделированной системы. В любой из N реализаций процесс наступления события A является случайной величиной, которая может приобретать значение $x_1 = 1$ с вероятностью p и $x_2 = 0$ с вероятностью $1-p$. Тогда можно найти математическое ожидание

$$E[\xi] = x_1 p + x_2 (1 - p) = p \quad (6.21)$$

и дисперсию

$$D[\xi] = (x_1 - E[\xi])^2 p + (x_2 - E[\xi])^2 (1 - p) = p(1 - p). \quad (6.22)$$

В качестве оценки p используют частоту наступления события A . Эта оценка несмещенная, состоятельная и эффективная.

При условии, что N заведомо задано, достаточно накапливать m :

$$\frac{m}{N} = \frac{1}{N} \sum_{i=1}^N \xi_i, \quad (6.23)$$

где ξ_i - наступление события A в реализации, $\xi_i = \{1, 0\}$.

Тогда находим

$$E\left[\frac{m}{N}\right] = p, \quad D\left[\frac{m}{N}\right] = \frac{p(1-p)}{N-1}. \quad (6.24)$$

В соответствии с центральной предельной теоремой (в данном случае можно взять теорему Лапласа) случайная величина $\frac{m}{N}$ будет распределение, близкое к нормальному (рис. 1). Поэтому каждой достоверности α из таблиц нормального распределения можно найти такую величину t_α , что точность ε будет равняться величине

$$\varepsilon = t_{\alpha} \sqrt{D \left[\frac{m}{N} \right]}. \quad (6.25)$$

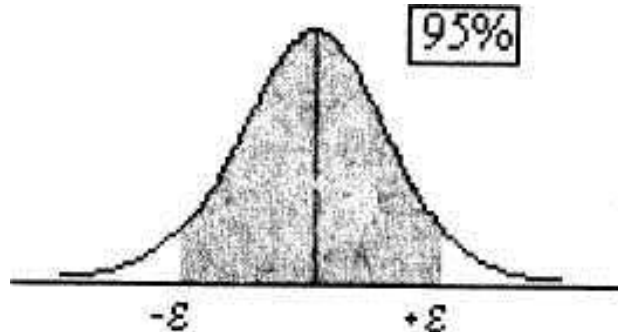


Рис.1. График нормального распределения

При $\alpha = 0.95$, $t_{\alpha} = 1.96$.

При $\alpha = 0.997$, $t_{\alpha} = 3$.

Подставим в уравнение (6.25) выражение дисперсии (6.245) получим:

$$\varepsilon = t_{\alpha} \sqrt{\frac{p(1-p)}{N-1}}. \quad (6.26)$$

Отсюда находим

$$N = t_{\alpha}^2 \frac{p(1-p)}{\varepsilon^2} + 1. \quad (6.27)$$

Поскольку вероятность p заранее неизвестна, прибегают к пробным испытаниям ($N = 50 \dots 100$), получают частоту $\frac{m}{N}$ и подставляют ее значения в выражение (6.27) вместо p , после чего определяют конечное количество испытаний.

Определение количества реализаций для оценки среднего значения случайной величины. Пусть случайная величина имеет математическое ожидание a и дисперсию σ^2 . В реализации с номером i она принимает значение x_i . Для оценки математического ожидания a используем среднее

$$\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i. \quad (6.28)$$

В соответствии с центральной предельной теоремой при больших значениях N среднее арифметическое \bar{x} будет нормально распределено с математическим ожиданием a и дисперсией $\frac{\sigma^2}{N-1}$. Тогда

$$\varepsilon = t_\alpha \frac{\sigma}{\sqrt{N-1}}. \quad (6.29)$$

Отсюда

$$N = \frac{t_\alpha^2 \sigma^2}{\varepsilon^2} + 1. \quad (6.30)$$

Поскольку дисперсия оцениваемой случайной величины неизвестна, необходимо провести 50-100 испытаний и оценить σ^2 , а потом полученное значение оценки подставить в формулу (6.30), чтобы определить необходимое количество реализаций N .

6. ПРИМЕРЫ ВЫПОЛНЕНИЯ ЗАДАЧ

Пример 1. Молодой пилот-стажер, который осуществляет свой первый полет, оказался небрежным курсантом и практически ничего не помнил из того, чему его учили. От его пилотирования инструктор сразу же потерял сознание, и при посадке молодому пилоту пришлось полагаться лишь на себя. Вероятность того, что за одну попытку стажер благополучно завершит посадку самолета, составляет 0,1. Определите число заходов на посадку, которое сделает стажер для того, чтобы полет благополучно закончился для него и его инструктора (количество бензина в баках считать неограниченным).

Решение. Вероятность благополучной посадки $\{A\}$ равна 0,1, т.е. $P\{A\} = 0.1$. Генерируем случайную величину r_i в интервале от 0 до 1.

Если $r_i \leq 0.1$, то событие $\{A\}$ произошло, процесс моделирования прекращаем.

Если $r_i > 0.1$, то событие $\{A\}$ не произошло, процесс моделирования продолжаем, запоминая число генераций случайной величины $N = N + 1$.

Повторяя данный процесс k раз, получим среднее число заходов, которое сделает стажер для того, чтобы полет благополучно закончился для него и его инструктора

$$\bar{N} = \frac{1}{k} \sum_{i=1}^k N_i.$$

Очевидно, что при $k \rightarrow \infty$, среднее число заходов будет

$$\bar{N} \rightarrow \frac{1}{P\{A\}} = 10.$$

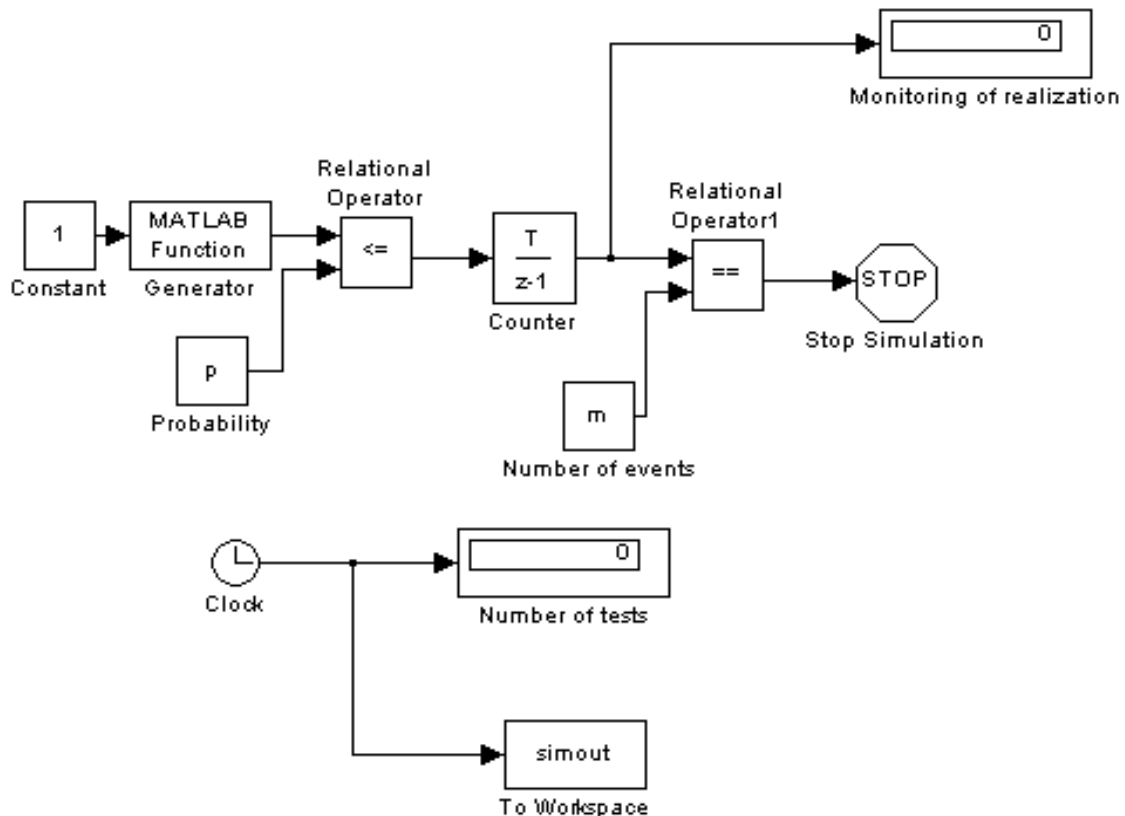
Поскольку реализовать процесс бесконечного приближения к истинному значению практически невозможно, задаются определенной точностью вычислений. Т.е. истинное значение математического ожидания искомой величины будет находиться в пределах интервала:

$$\bar{N} - t_{\alpha} \sqrt{\frac{\sigma^2}{k}} < E[N] < \bar{N} + t_{\alpha} \sqrt{\frac{\sigma^2}{k}} . \quad (6.31)$$

Значение дисперсии вычисляем по формуле (16), а для нахождения квантиля t_{α} используем функцию **norminv(p,m,s)**, где $p = (1-\alpha)/2$; α - достоверность, например, 0.95, $m = 0$ – математическое ожидание нормального распределения; $s = 1$ – дисперсия нормального распределения.

Для реализации задачи соберем S-модель, показанную на рисунке.

В блоке **MATLAB Function** введем формулу **unifrnd(0,u(1))**, представляющую собой генератор равномерно распределенного от 0 до 1 случайного числа. Использование данной функции продиктовано тем, что блок **Uniform Random Number** из библиотеки **Simulink** каждый раз генерирует одну и ту же последовательность случайных чисел, в то время как функция **unifrnd()**, каждый раз формирует новую последовательность случайных чисел. Назначение остальных блоков ясно из контекста. Для того чтобы сделать эту задачу более гибкой, вероятность события, p , и необходимое число его реализаций, m , задаем в файле-сценарии. Там же организуем цикл повторений работы с моделью и вычисления среднего числа заходов для благополучной посадки.



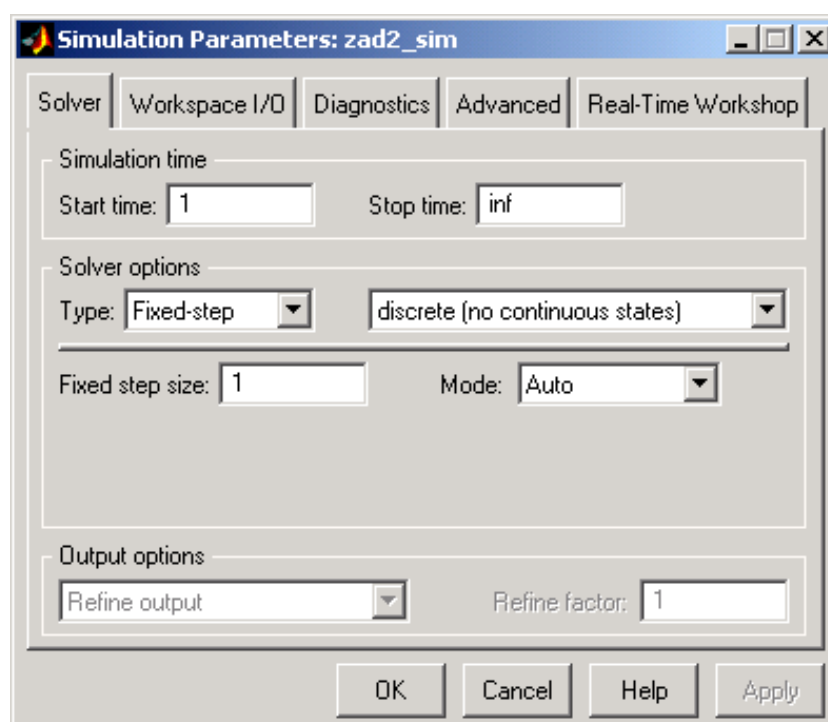
```

p=0.1; %вероятность наступления события
m=1; %число реализация события
k=500; %число испытаний
sum=0;
sum2=0;
load_system('zad2_sim')
for i=1:k
sim('zad2_sim')
n=simout(end,1); %текущее число заходов
sum=sum+n;
sum2=sum2+n^2;
end;
sr=sum/k %среднее число заходов
sigma2=1/(k-1)*(sum2-1/k*sum^2) %дисперсия числа заходов

% вычисление доверительного интервала
% пусть задана доверительность 0.95
% тогда уровень значимости равен 0.05
% для определения квантиля нормального распределения уровня
% 0.95
% используем обратную функцию нормального распределения
norminv(p,m,s)
% где p=(1-0.95)/2; m=0: s=1
t095=norminv(0.025,0,1);
% нижняя граница доверительного интервала
nmin=sr-t095*sqrt(sigma2/k)
% верхняя граница доверительного интервала
nmax=sr+t095*sqrt(sigma2/k)
% nmin < E[x] < nmax

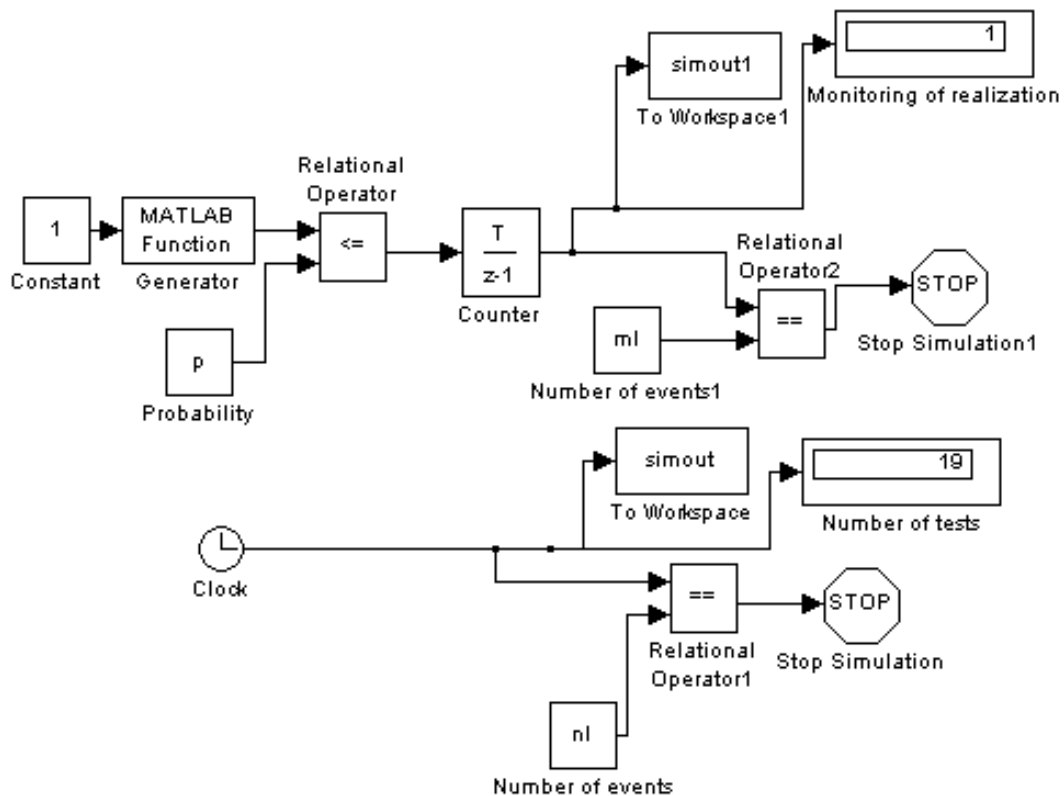
```

Параметры моделирования следует выставить так:



Пример 2. Молодой пилот-стажер, который полетел в свой первый полет и оказался небрежным курсантом. От его пилотирования инструктор сразу же потерял сознание, и при посадке молодому пилоту пришлось полагаться лишь на себя. Вероятность того, что за одну попытку стажер благополучно совершит посадку самолета, составляет 0,12. Найдите оценку вероятности того, что полет благополучно закончится для него и его инструктора, если бензина в баках хватает только на 20 кругов, а пилот-стажер для одной попытки посадки делает один круг.

Решение. Решение данной задачи во многом аналогично предыдущему примеру. Поэтому приведем здесь только реализацию задачи в Simulink



```

p=0.12; %вероятность наступления события
m1=1; %число реализация события
n1=20; %предельное число заходов
k=500; %число испытаний
sum=0;sum2=0;
load_system('zad8')
for i=1:k
sim('zad8')
m=simout1(end,1); %сдал или не сдал
n=simout(end,1); %текущее число заходов
sum=sum+m/n;
sum2=sum2+(m/n)^2;
end;
pr=sum/k %средняя вероятность сдачи за n1 заходов
sigma2=1/(k-1)*(sum2-1/k*sum^2) %дисперсия числа заходов

```

```

% вычисление доверительного интервала
% пусть задана доверительность 0.95
% тогда уровень значимости равен 0.05
% для определения квантиля нормального распределения уровня
0.95
% используем обратную функцию нормального распределения
norminv(p,m,s)
% где p=(1-0.95)/2; m=0: s=1
t095=norminv(0.025,0,1);
% нижняя граница доверительного интервала
nmin=pr-t095*sqrt(sigma2/k)
% верхняя граница доверительного интервала
nmax=pr+t095*sqrt(sigma2/k)
% nmin < E[x] < nmax

```

Параметры моделирования такие же, как и в примере 1.

7. ЗАДАЧИ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

Составьте S-модель и организуйте сценарий для решения ниже приведенных задач. При оформлении работ не забудьте описать общий ход решения (алгоритм).

Задача 1. Оцените значения константы π . Учитывая то, что площадь круга $S = \pi R^2$, используя метод Монте-Карло, найдите площадь круга, а через нее - значение π .

Задача 2. Вероятность получения зачета студентом, который не посещал лекционные и практические занятия, составляет p . Найдите оценку вероятности того, что студент получит зачет, если общее число сдач зачета не может превышать m .

№	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
p	0.13	0.22	0.07	0.35	0.52	0.14	0.19	0.63	0.5	0.1	0.24	0.82	0.9	0.32	0.41	0.21
m	7	7	10	3	3	7	7	3	3	10	8	2	3	5	10	10

Задача 3. Для получения допуска к экзамену студенту необходимо получить зачет по лабораторным работам (лабораторный курс состоит из K лабораторных работ). Вероятность зачета одной лабораторной работы для студента, который не защищал их в срок, составляет p за один «заход». Найдите оценку количества дней, нужных студенту для получения допуска к экзамену, если за один день он сможет получить зачет не более чем по одной лабораторной работе (на протяжении семестра он не защитил ни одной работы).

№	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
p	0.13	0.22	0.07	0.35	0.52	0.14	0.19	0.63	0.5	0.1	0.24	0.82	0.9	0.32	0.41	0.21
K	7	7	10	3	3	7	7	3	3	10	8	2	3	5	10	10

Задача 4. Стрелку необходимо выбить Z мишеней. Вероятность поражения одной мишени составляет p . Найдите оценку количества патронов, которое необходимо стрелку для того, чтобы попасть во все Z мишеней.

№	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
p	0.14	0.15	0.16	0.17	0.18	0.19	0.20	0.21	0.22	0.23	0.24	0.25	0.26	0.27	0.28	0.5
Z	4	4	4	4	4	5	5	5	5	5	6	6	6	6	6	10

8. ВОПРОСЫ ДЛЯ САМОКОНТРОЛЯ ЗНАНИЙ

1. В каком случае используется метод моделирования, называемый методом статистических испытаний?
2. В чем заключается суть метода статистических испытаний?
3. Опишите алгоритм метода статистических испытаний.
4. Какие способы реализации генераторов случайных чисел Вы знаете?
5. В чем заключается мультипликативный конгруэнтный метод?
6. Опишите способ моделирования случайных событий?
7. Каким образом моделируются группы несовместных событий?
8. Как моделируется случайная дискретная величина?
9. Что такое метод обратной функции, и как он используется при моделировании непрерывных случайных величин?
10. Запишите формулу нахождения оценки математического ожидания случайной величины.
11. Запишите формулу оценки дисперсии случайной величины.
12. Каким образом определить количество реализаций при моделировании случайной величины с необходимой точностью?
13. Как определить количество реализаций для оценки вероятности наступления события?
14. Что такое частота наступления события?
15. Как определить количество реализации для оценки среднего значения случайной величины?
16. Какие блоки Simulink используются для генерации случайных чисел?
17. В чем особенность работы таких блоков?
18. В чем отличие использования функций Matlab по сравнению со стандартными генераторами случайных чисел Simulink?
19. Что такое доверительность?
20. Каким образом доверительность связана с уровнем значимости?
21. Какая функция Matlab используется для определения квантиля нормального распределения?

ЛАБОРАТОРНАЯ РАБОТА № 7. НЕПРЕРЫВНО-СТОХАСТИЧЕСКИЕ МОДЕЛИ. Q-СХЕМЫ

ЦЕЛЬ РАБОТЫ: освоить технологию построения и исследования моделей систем массового обслуживания (с отказами) с помощью *Simulink* и *Stateflow*.

1. ОДНОКАНАЛЬНАЯ МОДЕЛЬ (С ОТКАЗАМИ) М/М/1

Одноканальная модель (с отказами) с пуассоновским входным потоком и экспоненциальным распределением длительности обслуживания.

Простейшей одноканальной моделью с вероятностным входным потоком и процедурой обслуживания является модель, характеризующаяся показательным распределением как длительностей интервалов между поступлениями требований, так и длительностей обслуживания. При этом плотность распределения длительностей интервалов между поступлениями требований имеет вид

$$f_1(t) = \lambda e^{(-\lambda t)},$$

где λ - интенсивность поступления заявок в систему.
Под интенсивностью потока понимают

$$\lambda(t) = \lim_{\tau \rightarrow 0} \frac{m(t, t + \tau)}{\tau},$$

где $m(t, t+t)$ - среднее число событий в интервале $(t, t+t)$.
Плотность распределения длительностей обслуживания:

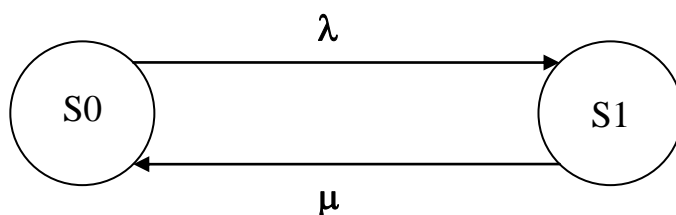
$$f_2(t) = \mu e^{(-\mu t)},$$

где μ - интенсивность обслуживания.

Поток заявок и обслуживания простейшие, т.е. обладающие свойствами стационарности (среднее число событий, воздействующих на систему, в течение единицы времени, остается постоянным), ординарности (вероятность попадания на элементарный участок времени двух и более событий пренебрежимо мала) и отсутствия последействия (для любых непересекающихся участков времени количество событий, попадающих на один из них, не зависит от того, сколько событий попало на другие участки времени).

Для простейшего потока интенсивность $\lambda = \text{const}$.

Пусть система работает с отказами. Необходимо определить абсолютную и относительную пропускную способности системы. Система имеет два состояния: S_0 - канал свободен и S_1 - канал занят.



Обозначим вероятности состояний:

$P_0(t)$ - вероятность состояния S_0 , $P_1(t)$ - вероятность состояния S_1 . Составим систему уравнений Колмогорова:

$$\frac{\partial}{\partial t} P_0(t) = -\lambda P_0(t) + \mu P_1(t)$$

$$\frac{\partial}{\partial t} P_1(t) = -\mu P_1(t) + \lambda P_0(t)$$

С учетом того, что $P_0(t) + P_1(t) = 1$, решение системы такое:

$$P_0(t) = \frac{\lambda e^{-(\lambda+\mu)t}}{\lambda + \mu} + \frac{\mu}{\lambda + \mu};$$

$$P_1(t) = 1 - P_0(t).$$

Для 1-канальной СМО с отказами вероятность $P_0(t)$ есть не что иное, как относительная пропускная способность системы q : $q = P_0(t)$.

По истечении большого интервала времени (при $t \rightarrow \infty$) достигается стационарный режим:

$$P_0 = \frac{\mu}{\lambda + \mu}.$$

Абсолютная пропускная способность (A) - среднее число заявок, которое может обслужить СМО в единицу времени:

$$A = \lambda P_0$$

или

$$A = \frac{\lambda \mu}{\lambda + \mu}.$$

Вероятность отказа в обслуживании заявки будет равна вероятности состояния "канал занят":

$$P_{отк} = 1 - P_0.$$

Данная величина может быть интерпретирована как средняя доля не обслуженных заявок среди поданных.

2. МНОГОКАНАЛЬНАЯ МОДЕЛЬ (С ОТКАЗАМИ) М/М/Н

Многоканальная модель (с отказами) с пуассоновским входным потоком и экспоненциальным распределением длительности обслуживания.

Пусть СМО с отказами имеет n каналов обслуживания, функционирующих независимо друг от друга. Входной поток заявок и поток обслуживания заявок являются пуассоновскими. Интенсивность поступления заявок равна λ , интенсивность обслуживания μ . Обозначим состояния СМО так:

S_0 - все каналы свободны;

S_1 - занят один канал, остальные свободны;

.....

S_k - заняты ровно k каналов, остальные свободны;

.....

S_n - все n каналов заняты, поступившая в СМО заявка получает отказ в обслуживании.

Уравнения Колмогорова для вероятностей состояний системы P_0, P_1, \dots, P_n будут иметь следующий вид:

$$\frac{\partial}{\partial t} P_0 = \lambda P_0 + \mu P_1,$$

.....

$$\frac{\partial}{\partial t} P_k = \lambda P_{k-1} - (\lambda + k\mu)P_k + \mu(k+1)P_{k+1}, \quad 1 \leq k \leq n-1,$$

.....

$$\frac{\partial}{\partial t} P_n = \lambda P_{n-1} - \mu n P_n.$$

Начальные условия решения СМО таковы: $P_0(0) = 1, P_1(0) = 0, \dots, P_n(0) = 0$.

Стационарное решение СМО имеет вид (формулы Эрланга):

$$P_k = \frac{\rho^k P_0}{k!}, \quad k=1, 2, \dots, n,$$

$$P_0 = \frac{1}{\sum_{k=0}^n \frac{\rho^k}{k!}},$$

где $\rho = \frac{\lambda}{\mu}$.

Вероятностные характеристики многоканальной СМО с отказами в стационарном режиме таковы:

1) вероятность отказа

$$P_{otk} = P_n;$$

2) вероятность того, что заявка будет принята к обслуживанию (относительная пропускная способность)

$$q = 1 - P_{otk};$$

3) абсолютная пропускная способность

$$A = \lambda q;$$

4) среднее число каналов, занятых обслуживанием

$$k_{sr} = \rho(1 - P_{otk}).$$

3. ПРИМЕР РЕШЕНИЯ

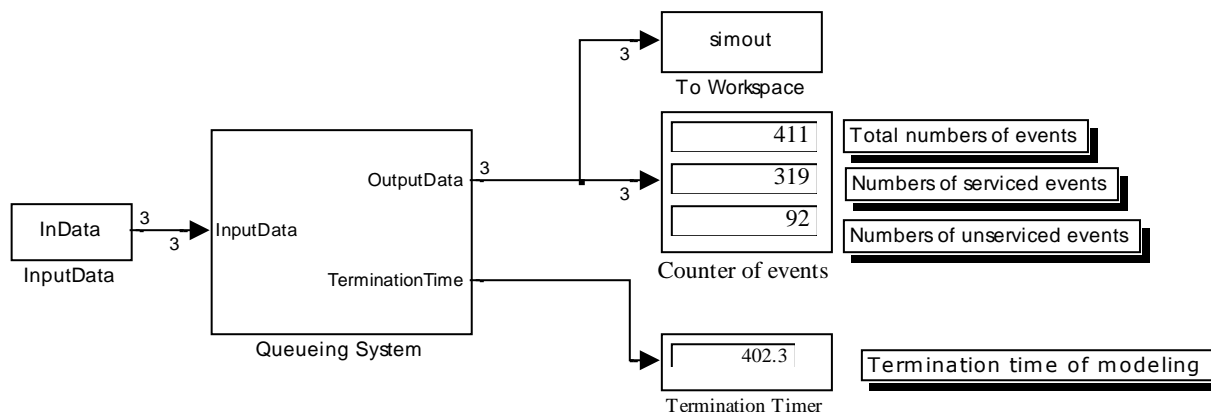
Пусть 3-канальная СМО представляет собой вычислительный центр (ВЦ) с тремя взаимозаменяемыми ПЭВМ для решения поступающих задач. Поток задач, поступающих на ВЦ, имеет интенсивность $\lambda = 1$ задачу в час. Средняя продолжительность обслуживания $1/\mu = 1.8$. Поток заявок на решение задач и поток обслуживания этих заявок являются простейшими. Вычислите финальные значения:

- вероятности состояний ВЦ;
- вероятности отказа в обслуживании заявки;
- относительной пропускной способности ВЦ;
- абсолютной пропускной способности ВЦ;
- среднего числа занятых ПЭВМ на ВЦ.

Остановимся немного подробнее на S-модели **CQS.mdl**.

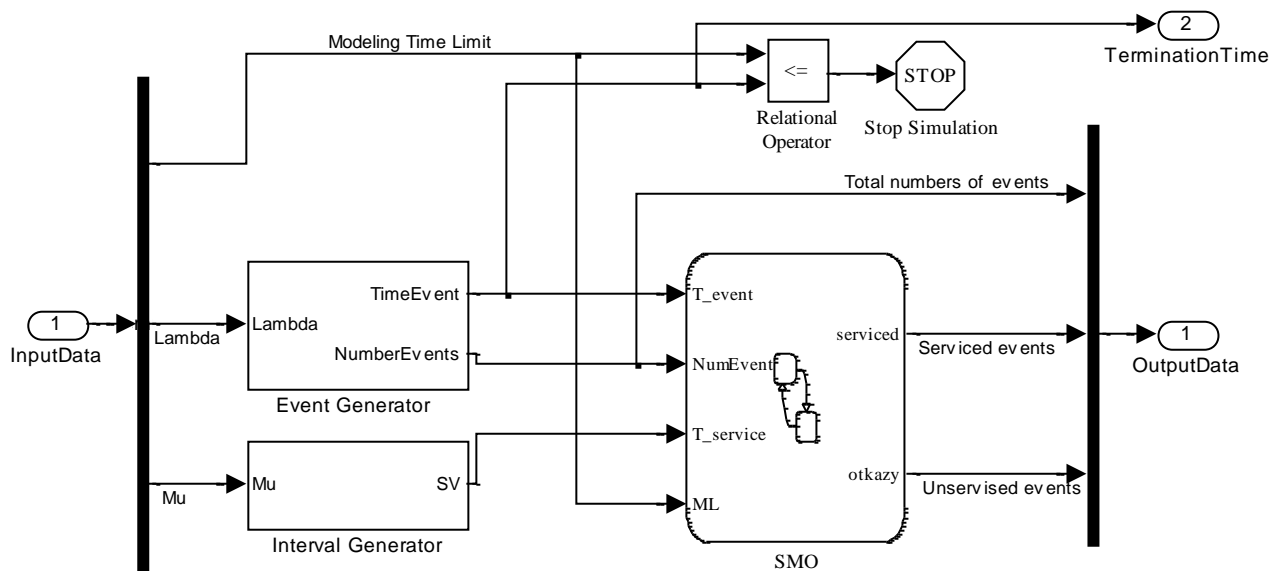
Система представляет трехканальную СМО с отказами, реализующую простейший пуассоновский поток заявок и периоды обслуживания заявок, распределенных по экспоненциальному закону.

Общая схема модели представлена ниже:



Блок **InputData** задает необходимые данные для модели (время окончания моделирования, λ и μ). Данные задаются посредством векторной константы, определяемой в файле-сценарии, **InData** = [**ML**; **lambda**; **mu**], где **ML** - время окончания моделирования; **lambda** = λ ; **mu** = μ .

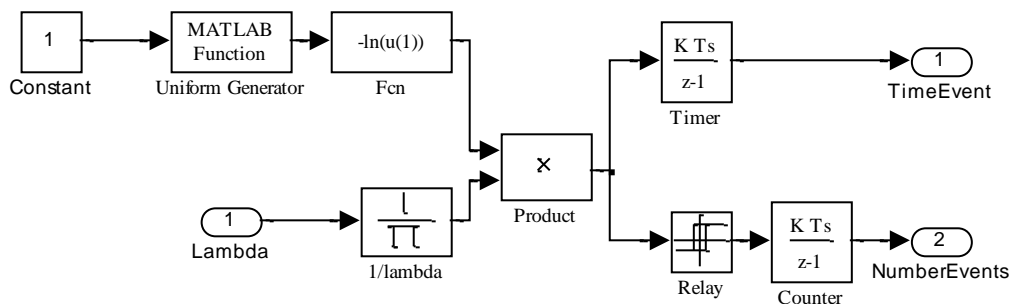
Блок **Queueing System** выполнен в виде подсистемы вида:



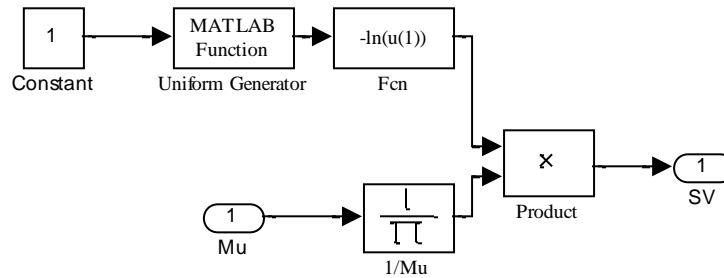
Данная подсистема в свою очередь состоит из 3-х подсистем: **Event Generator** – генератор заявок на обслуживание; **Interval Generator** – генератор интервалов обслуживания; **SMO** – **StateFlow** диаграмма.

Event Generator в свою очередь имеет следующую структуру: блок **Constant** не имеет никакого значения, а играет роль заглушки. Блок **MATLAB Function** играет роль генератора равномерно распределенного случайного числа - **unifrnd(0,1)**. Использование данного блока необходимо для реализации при каждом испытании модели новой последовательности СЧ. Блоки **Fcn**, **Product**, **1/lambda** реализуют формулу $x_i = -\frac{1}{\lambda} \ln \xi_i$.

Блок **Timer** продвигает модельное время по формуле: $T_{i+1} = T_i + x_i$. Группа блоков **Relay** и **Counter** реагируют на произошедшее событие и подсчитывают их количество.



Interval Generator действует аналогично и служит для формирования периода обслуживания заявки:



StateFlow диаграмма – **SMO** – представляет собой реализацию собственно СМО через концепцию теории конечных автоматов.

Здесь **s0** символизирует состояние отказов, **s1**, **s2**, **s3** – соответственно работу приборов обслуживания 1, 2 и 3. Рассмотрим состояния более подробно.

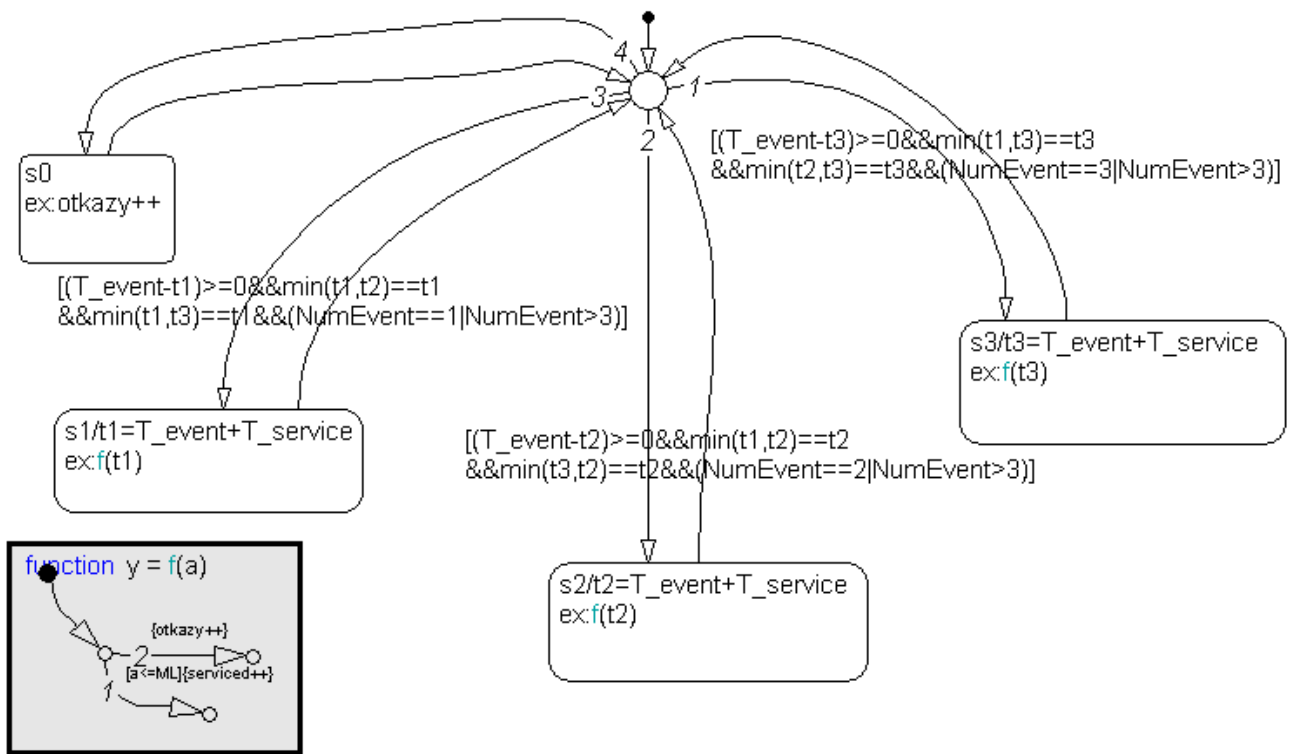
Состояние отказов **S0** – при выходе из состояния происходит суммирование отказов в обслуживании в выходной переменной **otkazy**.

Канал **S1** – при входе в данное состояние определяется время окончания обслуживания $t1 = T_event + T_service$. Текущее значение переменной **T_event** задается в блоке **Event Generator**, а текущее значение переменной **T_service** в блоке **Interval Generator**. При выходе из прибора обслуживания или состояния подсчитывается с помощью графической функции $y = f(a)$ количество обслуженных заявок в выходной переменной **serviced**. Действия остальных каналов аналогичны.

Условия перехода в то или иное состояние определяется через выполнения условия перехода для каждого состояния. Рассмотрим условие перехода в состояние **S1**:

$[(T_event - t1) \geq 0 \& \& \min(t1, t2) == t1 \& \& \min(t1, t3) == t1 \& \& (NumEvent == 1 | NumEvent > 3)]$

– если время наступления очередного события (заявки) оказывается больше или равно времени окончания обслуживания заявки в данном приборе (состоянии) и время окончания обслуживания в данном приборе оказывается минимальным (то есть предусматривается ситуация, что к моменту поступления новой заявки могут быть свободны все каналы, тогда занимается тот, который оказался свободным раньше) и при этом поступившее событие является либо первым, либо его номер больше 3, то занимается именно канал номер 1.



Остальные условия переходов в состояния **S2** и **S3** аналогичны с необходимыми изменениями для каждого канала.

Для состояния **S2**:

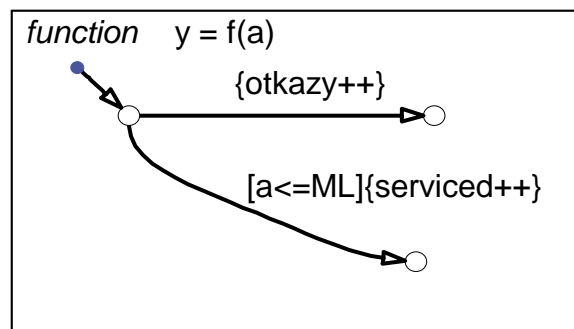
$[(T_event-t2) \geq 0 \&\& \min(t1, t2) == t2 \&\& \min(t3, t2) == t2 \&\& (NumEvent == 2 | NumEvent > 3)]$.

Для состояния **S3**:

$[(T_event-t3) \geq 0 \&\& \min(t1, t3) == t3 \&\& \min(t2, t3) == t3 \&\& (NumEvent == 3 | NumEvent > 3)]$

Если не одно из трех условий не выполняется, то система переходит в состояние отказа **S0**.

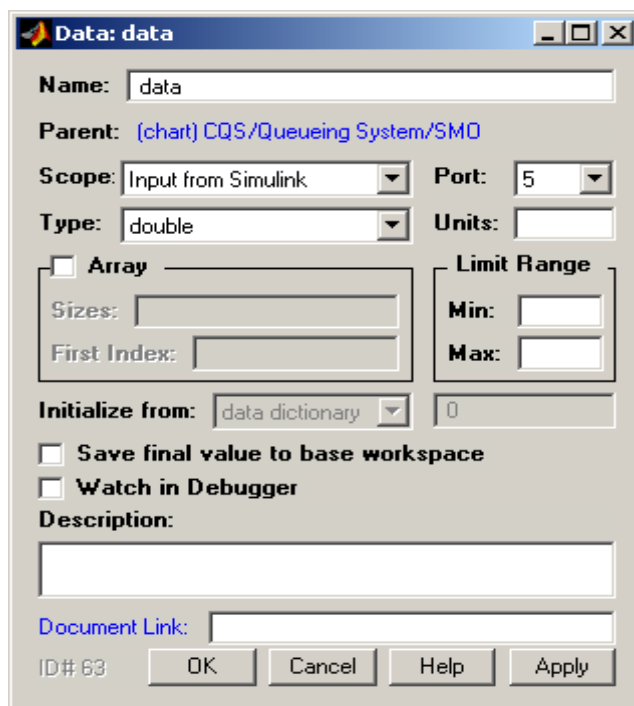
Как говорилось выше, в каждом блоке используется графическая функция $y = f(a)$, имеющая вид:



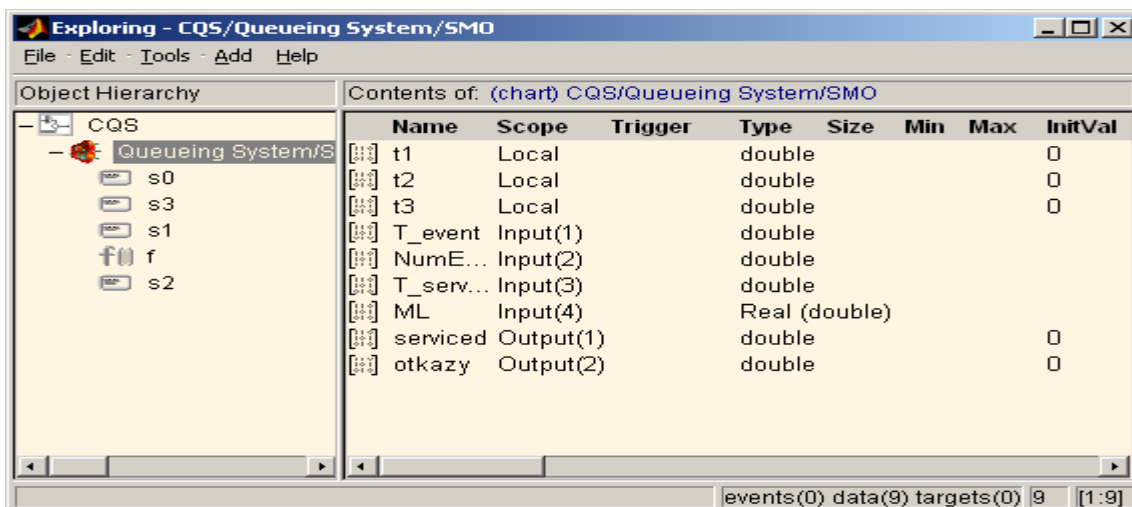
Как можно понять из рисунка, если время окончания обслуживания заявки в канале превышает предельное время моделирования, определяемое переменной **ML**, то заявка регистрируется как не обслуженная и накапливается в

переменной **otkazy**, иначе заявка регистрируется как обслуженная и накапливается в переменной **serviced**.

Для того чтобы система, описанная в **Stateflow**, могла обмениваться информацией с S-моделью, определяется список входных, выходных и внутренних переменных. Регистрация переменных осуществляется следующим образом: в меню выбирается команда **Add/Data** и далее выбирается тип переменной: **Local** – локальная; **Input from Simulink** – входная; **Output to Simulink** – выходная. При этом открывается диалоговое окно вида:



где и определяются все параметры данной переменной: имя, тип и т.п. Полный список используемых переменных можно найти в диалоговом окне **Explore**, если выполнить команду меню **Tools/Explore**:



Теперь рассмотрим, как работает данная модель. Формируем файл сценария, например, такого вида:

```

%Формирование исходных данных для моделирования
ML=4000;
lambda=1;
mu=1/1.8;
InData=[ML;lambda;mu];
%Задаем количество прогонов модели
k=10;
st=0;ss=0;so=0;
%Загружаем модель
load_system('CQS')
%Цикл испытаний
for i=1:k
sim('CQS')
%Сумма поступивших заявок во всех прогонах
st=st+simout(end,1);
%Сумма обслуженных заявок во всех прогонах
ss=ss+simout(end,2);
%Сумма необслуженных заявок во всех прогонах
so=so+simout(end,3);
end;
%Рассчитываем средние значения
st=st/k;ss=ss/k;so=so/k
%Рассчитываем вероятность отказов
Potk=so/st
%Относительная пропускная способность
q=1-Potk
%Абсолютная пропускная способность
A=lambda*q
%Среднее число занятых каналов
n=lambda/mu*q

```

Запускаем сценарий и по окончании работы модели в командном окне MATLAB получаем статистические данные нашей модели.

4. ЗАДАНИЕ

№	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
λ	0.95	0.5	10	1.5	4	40	0.5	0.7	2	1	30	3	0.1	0.75	1.5	20
$1/\mu$	1	1.5	0.2	3	0.3	1/30	1.2	3	0.8	3	4/60	0.5	8	1.2	2	0.4

1. Согласно варианту задания выполнить решение поставленной задачи следующими способами:
 - 1.1. Решить задачу, используя формулы Эрланга. (Решение задачи можно сделать в MS Excel либо другим доступным способом).
 - 1.2. Решить задачу, используя концепции Марковских цепей размножения и гибели, т.е. через уравнения Колмогорова. Решение задачи выполнить в Simulink.
 - 1.3. Решить задачу методом статистических испытаний, составив имитационную модель в среде Simulink.

2. Результаты вычислений оформить в виде отчета с подробным описанием работы каждой модели, используемых формул и с необходимым графическим материалом.

Пусть 3-канальная СМО представляет собой вычислительный центр (ВЦ) с тремя взаимозаменяемыми ПЭВМ для решения поступающих задач. Поток задач, поступающих на ВЦ, имеет интенсивность λ . Средняя продолжительность обслуживания $1/\mu$. Поток заявок на решение задач и поток обслуживания этих заявок являются простейшими. Вычислите финальные значения:

- вероятности состояний ВЦ;
- вероятности отказа в обслуживании заявки;
- относительной пропускной способности ВЦ;
- абсолютной пропускной способности ВЦ;
- среднего числа занятых ПЭВМ на ВЦ;
- сделайте вывод об эффективности использования данной СМО, если относительная пропускная способность должна быть не меньше 95%. В случае невыполнения этого условия сделайте необходимые изменения в данной СМО и выполните необходимые расчеты (расчеты выполнять только с помощью формул Эрланга)

5. ВОПРОСЫ ДЛЯ САМОКОНТРОЛЯ ЗНАНИЙ

1. Для изучения каких задач используется теория систем массового обслуживания?
2. Какие виды систем массового обслуживания различают?
3. Что такое входной поток заявок, очередь, устройство обслуживания?
4. Для каких типов СМО существуют аналитические решения?
5. Какие статистические показатели характеризуют СМО?
6. При каких условиях СМО достигает стационарного состояния?
7. Сформулируете закон Литтла для СМО?

ЛАБОРАТОРНАЯ РАБОТА №8.

РАСПОЗНАВАНИЕ ОБРАЗОВ С ОБУЧЕНИЕМ. ЛИНЕЙНЫЙ ДИСКРИМИНАНТНЫЙ АНАЛИЗ В СИСТЕМЕ STATISTICA

ЦЕЛЬ РАБОТЫ: провести анализ дискриминантной функции, наилучшим образом сепарирующей группы (классы) каких-либо объектов, используя метод линейного дискриминантного анализа в среде пакета **Statistica**.

1. ТЕОРЕТИЧЕСКОЕ ВВЕДЕНИЕ

Основная цель. Дискриминантный анализ используется для принятия решения о том, какие переменные различают (дискриминируют) две или более возникающие совокупности (группы). Например, некий исследователь в области образования предполагает исследовать, какие переменные относят выпускника средней школы к одной из трех категорий: (1) поступающий в колледж, (2) поступающий в профессиональную школу или (3) отказывающийся от дальнейшего образования или профессиональной подготовки. Для этой цели исследователь может собрать данные о различных переменных, связанных с учащимися школы. После выпуска большинство учащихся естественно должно попасть в одну из названных категорий. Затем можно использовать дискриминантный анализ для определения того, какие переменные дают наилучшее предсказание выбора учащимися дальнейшего пути.

Медик может регистрировать различные переменные, относящиеся к состоянию больного. Чтобы выяснить, какие переменные лучше предсказывают, что пациент, вероятно, выздоровел полностью (группа 1), частично (группа 2) или совсем не выздоровел (группа 3), используют дискриминантный анализ. Биолог может записать различные характеристики сходных типов (групп) цветов, чтобы затем провести анализ дискриминантной функции, наилучшим образом разделяющей типы или группы.

Функция дискриминации для двух групп. Для двух групп часто используется *линейный дискриминантный анализ Фишера* (Fisher, 1936), который может рассматриваться как процедура множественной регрессии. (С вычислительной точки зрения все эти подходы аналогичны). Если вы кодируете две группы как 1 и 2, и затем используете эти переменные в качестве зависимых переменных в множественной регрессии, то получите результаты, аналогичные тем, которые получили бы с помощью *Дискриминантного анализа*. В общем, в случае двух совокупностей вы подгоняете линейное уравнение следующего типа:

$$\text{Группа} = a + b_1 \cdot x_1 + b_2 \cdot x_2 + \dots + b_m \cdot x_m,$$

где a является константой, и $b_1 \dots b_m$ являются коэффициентами регрессии. Переменные с наибольшими регрессионными коэффициентами вносят наибольший вклад в дискриминацию.

Функция дискриминации для нескольких групп. Если имеется более двух групп, то можно оценить более, чем одну дискриминантную функцию подобно тому, как это было сделано ранее. Например, когда имеются три совокупности, вы можете оценить: (1) - функцию для дискриминации между совокупностью 1 и совокупностями 2 и 3, взятыми вместе, и (2) - другую функцию для дискриминации между совокупностью 2 и совокупностью 3. Например, вы можете иметь одну функцию, дискриминирующую между теми выпускниками средней школы, которые идут в колледж, против тех, кто этого не делает (но хочет получить работу или пойти в училище), и вторую функцию для дискриминации между теми выпускниками, которые хотят получить работу против тех, кто хочет пойти в училище. Коэффициенты b в этих дискриминирующих функциях могут быть проинтерпретированы тем же способом, что и ранее.

Вычислительный подход. Коэффициенты в дискриминирующих функциях могут определяться **стандартным методом** (регрессионный анализ с участием всех факторов) и **пошаговым методом** (пошаговый анализ с включением и пошаговый анализ с исключением дискриминирующих переменных).

Пошаговый анализ с включением. В пошаговом анализе дискриминантных функций модель дискриминации строится по шагам. Точнее, на каждом шаге просматриваются все переменные, и находится та из них, которая вносит наибольший вклад в различие между совокупностями. Эта переменная должна быть включена в модель на данном шаге, и происходит переход к следующему шагу.

Пошаговый анализ с исключением. Можно также двигаться в обратном направлении, в этом случае все переменные будут сначала включены в модель, а затем на каждом шаге будут устраняться переменные, вносящие малый вклад в предсказания. Тогда в качестве результата успешного анализа можно сохранить только "важные" переменные в модели, то есть те переменные, чей вклад в дискриминацию больше остальных.

F для включения, F для исключения. Пошаговая процедура "руководствуется" соответствующим значением F (критерий Фишера) для включения и соответствующим значением F для исключения. Значение F статистики для переменной указывает на ее статистическую значимость при дискриминации между совокупностями, то есть, она является мерой вклада переменной в предсказание членства в совокупности. Решение также может быть принято по уровню значимости, найденному по распределению Фишера.

Дискриминация. Как только модель установлена и получены дискриминирующие функции, возникает вопрос о том, как хорошо они могут *предсказывать*, к какой совокупности принадлежит конкретный образец?

Априорная и апостериорная классификация. Классификация действует лучшим образом для выборки, по которой была проведена оценка дискриминирующей функции (*апостериорная* классификация), чем для свежей выборки (*априорная* классификация). Трудности с *априорной* классификацией будущих

образцов заключается в том, что никто не знает, что может случиться. Намного легче классифицировать уже имеющиеся образцы. Поэтому оценивание качества процедуры классификации никогда не производят по той же самой выборке, по которой была оценена дискриминирующая функция. Если желают использовать процедуру для классификации будущих образцов, то ее следует "испытать" на новых объектах. **Можно использовать следующий прием: получить дискриминирующую функцию на части выборки (выполнить обучение), а на второй части выборки оценить качество классификации.**

Дискриминация наблюдений. Для каждого наблюдения рассчитывается значение дискриминирующей функции. В общем случае наблюдение считается принадлежащим той совокупности, для которой получено **наибольшее значение дискриминирующей функции.** Поэтому, если вы изучаете выбор карьеры или образования учащимися средней школы после выпуска (поступление в колледж, в профессиональную школу или получение работы) на основе нескольких переменных, полученных за год до выпуска, то можете использовать функции классификации, чтобы предсказать, что наиболее вероятно будет делать каждый учащийся после выпуска. Однако вы хотели бы определить *вероятность*, с которой учащийся сделает предсказанный выбор. Эти вероятности называются *апостериорными*, и их также можно вычислить. Однако для понимания, как эти вероятности вычисляются, вначале рассмотрим так называемое *расстояние Махаланобиса*.

Расстояние Махаланобиса. В общем, расстояние Махаланобиса является мерой расстояния между двумя точками в пространстве, определяемом двумя или более *коррелированными* переменными. Например, если имеются всего две некоррелированные переменные, то расстояние Махаланобиса между точками будет в этом случае равно расстоянию Евклида, т.е. расстоянию, измеренному, например, рулеткой. Если имеются три некоррелированные переменные, то для определения расстояния вы можете по-прежнему использовать рулетку (на 3М диаграмме). При наличии более трех переменных вы не можете более представить расстояние на диаграмме. Также и в случае, когда переменные коррелированы, то оси на графике могут рассматриваться как *неортогональные* (они уже не направлены под прямыми углами друг к другу). В этом случае простое определение расстояния Евклида не подходит, в то время как расстояние Махаланобиса является адекватно определенным в случае наличия корреляций.

Расстояние Махаланобиса и классификация. Для каждой совокупности в выборке вы можете определить положение точки, представляющей средние для всех переменных в многомерном пространстве, определенном переменными рассматриваемой модели. Эти точки называются *центроидами* группы. Для каждого наблюдения вы можете затем вычислить его расстояние Махаланобиса от каждого центроида группы. Вы признаете наблюдение принадлежащим к той группе, к которой он ближе, т.е. когда **расстояние Махаланобиса до нее минимально.**

Апостериорные вероятности классификации. Используя для классификации расстояние Махаланобиса, вы можете теперь получить вероятность того, что образец принадлежит к конкретной совокупности. Эти вероятности

называются *апостериорными* вероятностями. *Апостериорные* вероятности могут быть вычислены для всех наблюдений.

Априорные вероятности классификации. Имеется одно дополнительное обстоятельство, которое следует рассмотреть при классификации образцов. Иногда вы знаете заранее, что в одной из групп имеется больше наблюдений, чем в другой. Поэтому *априорные* вероятности того, что образец принадлежит такой группе, выше. Например, если вы знаете заранее, что 60% выпускников вашей средней школы обычно идут в колледж, (20% идут в профессиональные школы и остальные 20% идут работать), то вы можете уточнить предсказание таким образом: при всех других равных условиях более вероятно, что учащийся поступит в колледж, чем сделает два других выбора. Вы можете установить различные *априорные* вероятности, которые будут затем использоваться для уточнения результатов классификации наблюдений (и для вычисления *апостериорных* вероятностей).

На практике исследователю необходимо задать себе вопрос, является ли неодинаковое число наблюдений в различных совокупностях в первоначальной выборке отражением истинного распределения в популяции, или это только случайный результат процедуры выбора. В первом случае вы должны положить *априорные* вероятности пропорциональными объемам совокупностей в выборке; во втором - положить *априорные* вероятности одинаковыми для каждой совокупности. Спецификация различных *априорных* вероятностей может сильно влиять на точность классификации.

Рассмотрим процедуру дискриминантного анализа в пакете STATISTICA6 на примере решения конкретной задачи.

2. ЗАДАНИЕ № 1. КЛАССИЧЕСКИЙ ПРИМЕР ФИШЕРА

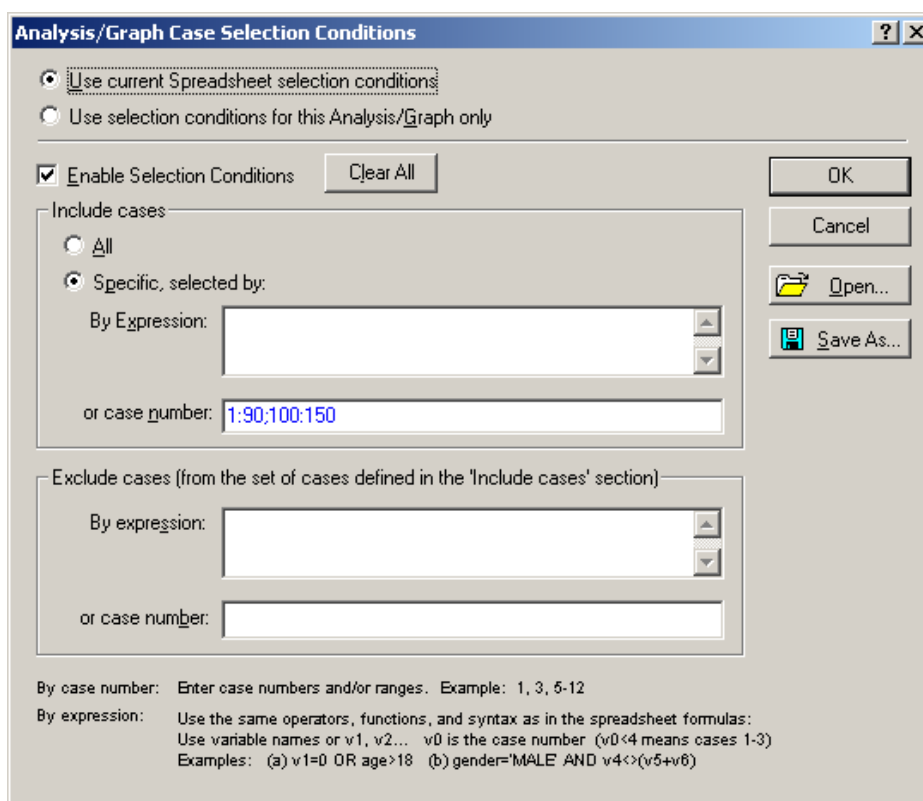
Воспользуемся данными классического примера Фишера. Задача состоит в том, чтобы по результатам измерений длины и ширины чашелистиков и лепестков цветков ириса отнести ирис к одному из трех типов: SETOSA, VERSICOL, VIRGINIC. Данные для этого примера имеются в файле Irisdat.sta (находится в папке Datasets папки Examples стандартной поставки пакета Statistica). В файле содержатся результаты измерений 150 цветков ириса, по 50 каждого типа.

1. Выполнить дискриминацию цветков ириса по части выборки (обучение), используя стандартный метод и один из пошаговых методов по варианту задания. (Forward stepwise – пошаговый с включением, Backward stepwise – пошаговый с исключением).
2. Проверить качество классификации на части выборки, не участвующей в выработке классификационных функций.
3. Классифицировать цветок ириса с параметрами: PETALLEN=5.2, PETALWID=1.8, SEPALLEN=6.5, SEPALWID=2.9

Вариант	1	2	3	4	5	6	7	8	9	10	11	12	13
Метод	For	Back	Back	For	Back	For	Back	For	For	Back	For	Back	For
Выборка для обучения (номера опытов)	11-150	1-140	1-100, 111-150	1-140	11-150	1-100, 111-150	1-120, 131-150	1-125, 136-150	1-120, 131-150	1-125, 136-150	6-145	6-145	3-142

1. Дискриминация с обучением.

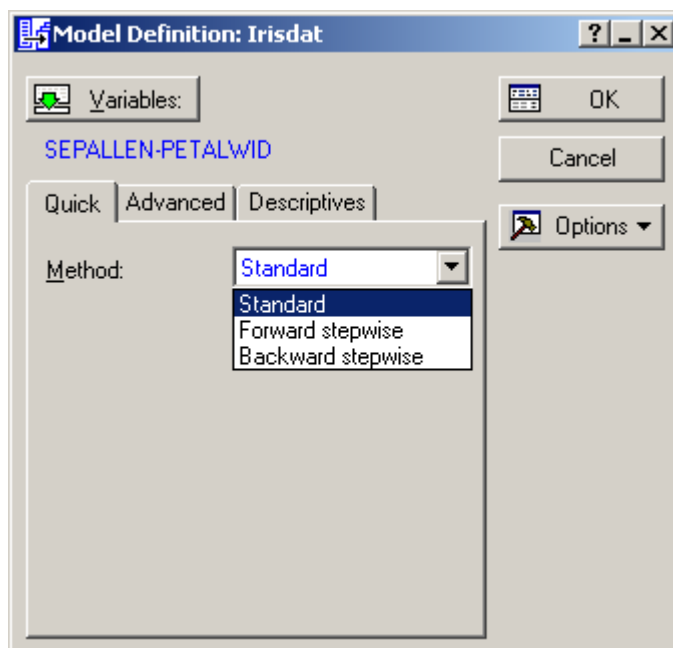
- Загрузить пакет и открыть файл Irisdat.sta. Добавить строку с номером 151 и ввести в неё значения переменных для цветка, который необходимо будет классифицировать (из п.3 ЗАДАЧИ).
- Вызвать модуль дискриминантного анализа: в меню команд открыть **Statistics**, выбрать **Multivariate Exploratory Techniques** и **Discriminant Analysis** – появится стартовая панель модуля.
- Задать объём обучающей выборки: щёлкнуть **Select cases** и указать номера опытов. Например, возьмём для обучения опыты 1-90 и 100-150 (следовательно, по опытам 91-99 будем в дальнейшем проверять качество дискриминации).



- Выбрать переменные для анализа: щёлкнуть **Variables (Переменные)** и в качестве **Группирующей переменной** — **Grouping variable** выбрать переменную **IRISTYPE** — тип ириса; в качестве **Независимых переменных** — **Independent variables** выбрать переменные **SEPALLEN** – длина

чашелистика, **SEPALWID** – ширина чашелистика, **PETALLEN** – длина лепестка, **PETALWID** — ширина лепестка.

5. Выбрать метод. В стартовой панели включить опцию **Advanced options (stepwise analysis)**, **OK**. В окне **Model Definition (Определение модели)** выбирается вычислительный метод: стандартный (**Standard**), пошаговый с включением (**Forward stepwise**) или пошаговый с исключением (**Backward stepwise**). Выберем решение стандартным методом.



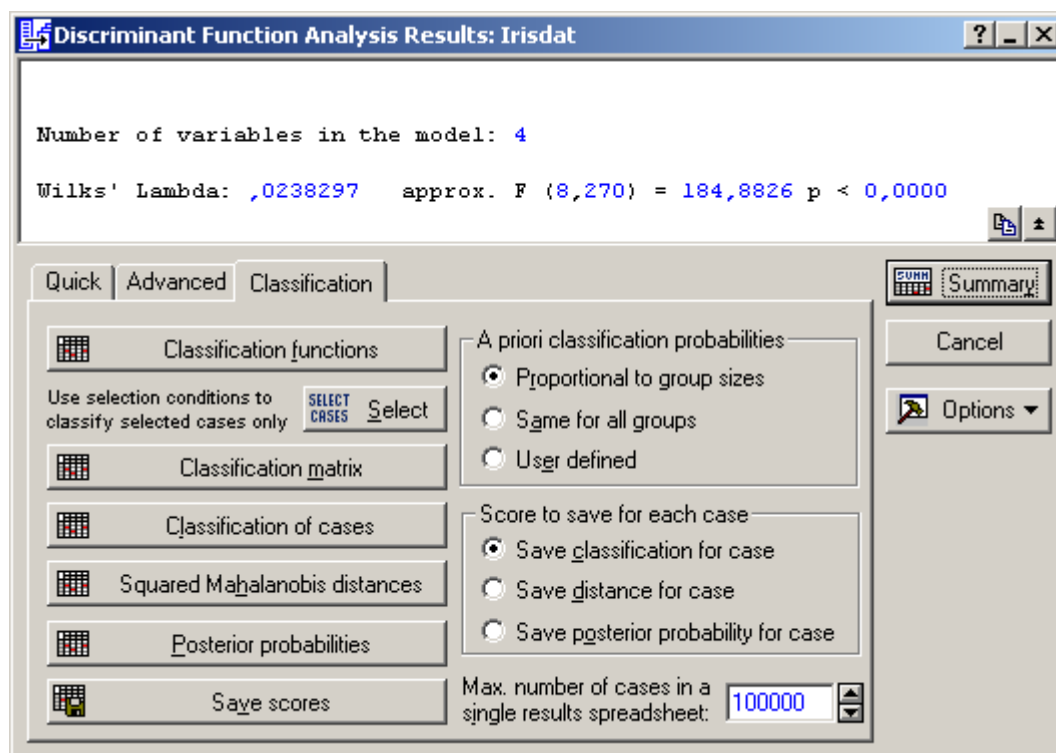
6. Вызвать просмотр описательных статистик: щёлкнуть закладку **Descriptives** и **Review descriptive statistics**. Щелчком по **Means & number of cases** на закладке **Within** открывается таблица средних значений дискриминирующих переменных в каждой группе и во всех группах.

	SEPALLEN	SEPALWID	PETALLEN	PETALWID	Valid N
SETOSA	5,010638	3,429787	1,457447	0,244681	47
VERSICOL	5,970455	2,781818	4,293182	1,336364	44
VIRGINIC	6,588000	2,974000	5,552000	2,026000	50
All Grps	5,869503	3,065958	3,794326	1,217021	141

Из таблицы видно, что средние значения переменных в группе **SETOSA** отличаются от средних в группах **VERSICOL** и **VIRGINIC** значительно сильнее, чем средние в группах **VERSICOL** и **VIRGINIC** между собой. Средние значения и доверительные интервалы для средних каждой переменной можно более наглядно увидеть на графике «ящик с усами». Для этого щёлкнуть **Box plot of means by group** и просмотреть все переменные по очереди. По графикам можно увидеть, что переменные

PETALLEN и **PETALWID** обладают большей дискриминирующей силой и сами по себе могут быть дискриминаторами.

5. Выполнить дискриминантный анализ и получить результаты. Вернуться в окно **Model Definition** и щелчком по ОК запустить процесс дискриминации. Результаты дискриминантного анализа в диалоговом окне **Discriminant Function Analysis Results**.



Информационная часть окна сообщает:

- Wilks lambda — значение лямбды Уилкса ;
- approx. F (8,270) – приближенное значение F-статистики, связанной с лямбдой Уилкса;
- p — уровень значимости F-критерия для значения.

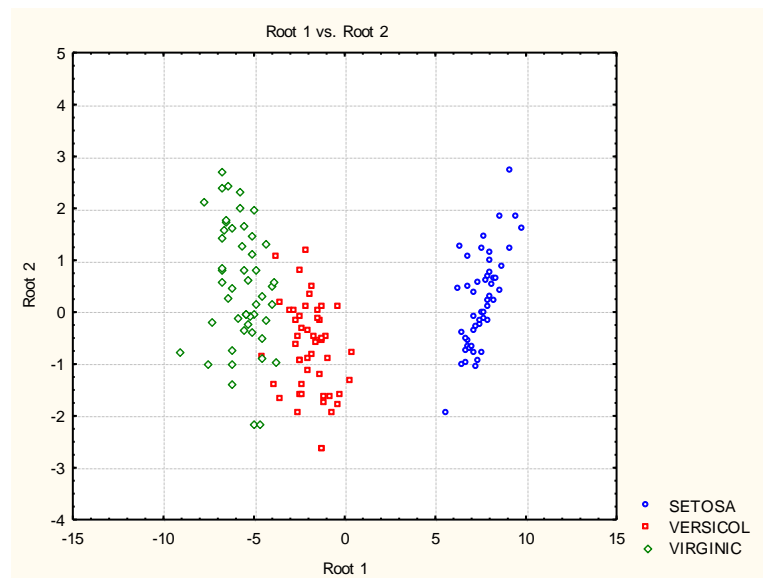
Значения статистики лямбда Уилкса лежат в интервале [0,1]. Значения, лежащие около 0, свидетельствуют о хорошей дискриминации. Значения статистики Уилкса, лежащие около 1, свидетельствуют о плохой дискриминации. Иными словами, это можно выразить следующим образом: если значение лямбда Уилкса близки к 0, то мощность дискриминации (мощность=1- вероятность ошибки) близка к 1, если лямбда Уилкса близка к 1, то мощность близка к 0. По уровню значимости проверяется значимость отличия лямбды Уилкса от нуля. Т.к. $p < 0.05$, лямбда Уилкса незначимо отличается от нуля и мощность дискриминации близка к 1.

Нажмите кнопку **Variables in the model (Переменные, включенные в модель)**. По таблице можно определить дискриминирующие свойства каждой переменной. Дискриминирующие свойства тем сильнее, чем меньше Partial Lambda, или чем больше F, или чем меньше рассчитанный по распределению Фишера уровень значимости (**p-level**).

	Wilks'	Partial	F-remove	p-level	Toler.	1-Toler.
SEPALLEN	0,025529	0,933447	4,81262	0,009573	0,359194	0,640806
SEPALWID	0,030505	0,781168	18,90907	0,000000	0,638786	0,361214
PETALLEN	0,036053	0,660968	34,62300	0,000000	0,378244	0,621756
PETALWID	0,032237	0,739196	23,81539	0,000000	0,678468	0,321532

Из таблицы видно, что все переменные обладают дискриминирующими свойствами, но дискриминирующие свойства переменной SEPALLEN несколько слабее.

Чтобы представить разделение элементов выборки на группы графически, в окне **Discriminant Function Analysis Results** открыть закладку **Advanced**, выбрать **Perform canonical analysis** и **Scatterplot of canonical scores**.



Из графика видно, что элементы группы SETOSA дискриминируются очень чётко. Границы групп VERSICOL и VIRGINIC перекрываются, поэтому элементы выборки на границе могут быть дискриминированы с ошибками.

Для вывода функции дискриминации необходимо вернуться в окно **Discriminant Function Analysis Results**, выбрать закладку **Classification** и нажать кнопку **Classification functions**. Появится окно, в котором приведены коэффициенты функции дискриминации для каждой группы.

	SETOSA	VERSICOL	VIRGINIC
SEPALLEN	23,3023	15,4255	12,054
SEPALWID	24,2525	8,7129	5,936

PETALLEN	-15,7485	5,4804	12,836
PETALWID	-16,6351	6,4877	20,591
Constant	-87,5573	-75,4311	-106,062

Функции дискриминации:

$$\text{SETOSA} = -15.75 * \text{PETALLEN} - 16.64 * \text{PETALWID} + 23.3 * \text{SEPALLEN} + 24.25 * \text{SEPALWID} - 87.56$$

$$\text{VERSICOL} = 5.48 * \text{PETALLEN} + 6.49 * \text{PETALWID} + 15.43 * \text{SEPALLEN} + 8.71 * \text{SEPALWID} - 75.43$$

$$\text{VIRGINIC} = 12.84 * \text{PETALLEN} + 20.59 * \text{PETALWID} + 12.05 * \text{SEPALLEN} + 5.94 * \text{SEPALWID} - 106.06$$

6. Оценить дискриминирующие свойства полученных классификационных функций на обучающей выборке. Щелчком по клавише **Classification of Cases** вызывается таблица, в которой отмечены неверно классифицированные объекты. Чтобы узнать в какой конкретно класс попали эти объекты, можно посмотреть таблицы с квадратом расстояния Махаланобиса и с апостериорными вероятностями. Щелчком по клавише **Classification matrix** выводится таблица, которая указывает процент корректной классификации в обучающей выборке.

	Percent	SETOSA	VERSICOL	VIRGINIC
SETOSA	100,0000	47	0	0
VERSICOL	95,4545	0	42	2
VIRGINIC	98,0000	0	1	49
Total	97,8723	47	43	51

Таким образом, полученная дискриминантная функция позволяет выполнить классификацию объектов с вероятностью 97.9%, причём цветки класса SETOSA классифицируются с вероятностью 100%. В классификации двух других групп наблюдается незначительный процент ошибки, что и следовало ожидать.

II. Оценка качества дискриминации на элементах выборки, не вошедших в обучение.

На закладке **Classification** щёлкнуть **Select cases** и указать номера соответствующих опытов, в нашем случае – это опыты с 91 до 99. Далее просмотреть все таблицы, указанные в п.8. В нашем случае все 9 опытов классифицируются с вероятностью 100%.

III. Классификация нового объекта.

На закладке **Classification** щёлкнуть **Select cases** и указать номер нового объекта 151. Посмотреть расстояние Махаланобиса:

	Observed	SETOSA	VERSICOL	VIRGINIC
151	---	143,4805	7,601916	1,780599

и апостериорную вероятность:

	Observed	SETOSA	VERSICOL	VIRGINIC
151	---	0,000000	0,045717	0,954283

По этим данным новый цветок следует отнести к классу **VIRGINIC**.

IV. Получить дискриминирующую функцию методом, указанным в индивидуальном задании. Что изменилось по сравнению со стандартным методом?

3. ЗАДАНИЕ № 2. МЕДИЦИНСКАЯ ЗАДАЧА

Имеются данные клиники, в которой зарегистрированы результаты обследования 103 человек с установленным диагнозом: группа 1 – гангренозный аппендицит (28 наблюдений), группа 2 – флегмонозный аппендицит (25 наблюдений), группа 3 – катаральный аппендицит (26 наблюдений) и группа 4 – неподтверждённый диагноз (24 наблюдения). Исходными признаками служили 8 симптомов. Данные и описание симптомов находятся в файле **appendix.sta**.

Выполнить дискриминантный анализ данных по всей выборке и с обучением. Оценить качество дискриминации с помощью полученной дискриминирующей функции.

4. ВОПРОСЫ ДЛЯ САМОКОНТРОЛЯ ЗНАНИЙ

1. Какова цель дискриминантного анализа?
2. Что такое дискриминирующая функция?
3. Перечислите методы расчёта коэффициентов в дискриминирующих функциях.
4. Суть метода пошагового анализа с включением.
5. Суть метода пошагового анализа с исключением.
6. В чём суть дискриминантного анализа с обучением?
7. Что такое априорная и апостериорная классификация?

8. Каким образом можно классифицировать новый объект, если известны дискриминирующие функции для всех групп?
9. Что такое расстояние Махаланобиса?
10. Каким образом можно классифицировать новый объект, используя расстояние Махаланобиса?
11. Априорная и апостериорная вероятности.
12. Какой модуль пакета Statistica используется для дискриминантного анализа и в каком разделе пакета он находится?
13. Что вводится в качестве **Variables** и что в качестве **Grouping variable** в задачах лабораторной работы?
14. Для каких величин строится и что характеризует график «ящик с усами»?
15. В каком интервале находятся значения статистики лямбда Уилкса?
16. Как определяется мощность дискриминации?
17. По каким статистическим показателям можно судить о силе дискриминирующих свойств переменных?
18. По какому показателю можно оценить дискриминирующие свойства полученных классификационных функций?

ЛАБОРАТОРНАЯ РАБОТА №9.

РЕШЕНИЕ ЗАДАЧ С ИСПОЛЬЗОВАНИЕМ НЕЙРОННЫХ СЕТЕЙ В *MATLAB*

ЦЕЛЬ РАБОТЫ: освоение технологии решения задач с использованием нейронных сетей в пакетах расширения *Neural Networks Toolbox* и *Simulink*.

1. ОБЩИЕ СВЕДЕНИЯ О НЕЙРОННЫХ СЕТЯХ

Нейронные сети (НС) можно рассматривать как современные вычислительные системы, которые преобразуют информацию по образу процессов, происходящих в мозгу человека. НС представляют собой простые математические модели мозгоподобных систем, функционирующих как параллельные распределённые вычислительные сети. Проследив историю развития НС, можно сказать, что учёных давно интересует создание компьютерных систем, имитирующих мозговую деятельность, т.е. обладающих интеллектом. Интеллект и мышление органически связаны с решением таких задач, как доказательство теорем, логический анализ, распознавание ситуаций, планирование поведения, игры и управление в условиях неопределённости. Характерными чертами интеллекта, проявляющимися в процессе решения задач, являются способность к обучению, обобщению, накоплению опыта (знаний и навыков) и адаптации к изменяющимся условиям в процессе решения задач. Благодаря этим качествам интеллекта мозг может решать разнообразные задачи, а также легко перестраиваться с решения одной задачи на другую. Таким образом, мозг, наделенный интеллектом, является универсальным средством решения широкого круга задач (в том числе неформализованных), для которых нет стандартных, заранее известных методов решения.

Существуют различные подходы к моделированию интеллекта. В частности, интересен план имитации мышления, предложенный А. Тьюрингом. "Пытаясь имитировать интеллект взрослого человека, — пишет Тьюринг, — мы вынуждены много размышлять о том процессе, в результате которого человеческий мозг достиг своего настоящего состояния. Почему бы нам вместо того, чтобы пытаться создать программу, имитирующую интеллект взрослого человека, не попытаться создать программу, которая имитировала бы интеллект ребенка? Ведь если интеллект ребенка получает соответствующее воспитание, он становится интеллектом взрослого человека. Наш расчет состоит в том, что устройство, ему подобное, может быть легко запрограммировано... Таким образом, мы расчленим нашу проблему на две части: на задачу построения "программы-ребенка" и задачу "воспитания" этой программы". Забегая вперед, можно сказать, что именно этот путь используют практически все системы НС.

В 1957г. американский физиолог Ф.Розенблатт предложил модель зрительного восприятия и распознавания — перцептрон. (Перцепция это способность живых существ к восприятию окружающей действительности с помощью органов чувств и нервной системы). Появление машины, способной обучаться

понятиям и распознавать предъявляемые объекты, оказалось чрезвычайно интересным не только физиологам, но и представителям других областей знания и породило большой поток теоретических и экспериментальных исследований. Можно сказать, что все НС «вышли из персептрона Розенблатта».

НС являются одной из составляющих систем искусственного интеллекта (ИИ). Аппарат ИИ используется при решении задач, когда невозможно получить полную информацию о системе или когда система достаточно сложная и невозможно описать её поведение детерминированными законами. Однако аппарат НС может использоваться и самостоятельно для решения задач, которые невозможно решить традиционными методами и для упрощения решения широкого круга задач по сравнению с традиционными математическими методами. Именно этот аспект использования НС мы и будем рассматривать.

1.1. Как работает биологическая нейронная сеть

Нервная система и мозг человека состоят из нейронов, соединенных между собой нервными волокнами. Нервные волокна способны передавать электрические импульсы между нейронами. Все процессы передачи раздражений от нашей кожи, ушей и глаз к мозгу, процессы мышления и управления действиями - все это реализовано в живом организме как передача электрических импульсов между нейронами. Рассмотрим строение биологического нейрона (рис.9.1). Каждый нейрон имеет отростки нервных волокон двух типов - дендриты, по которым принимаются импульсы, и единственный аксон, по которому нейрон может передавать импульс. Аксон контактирует с дендритами других нейронов через специальные образования - синапсы, которые влияют на силу импульса.

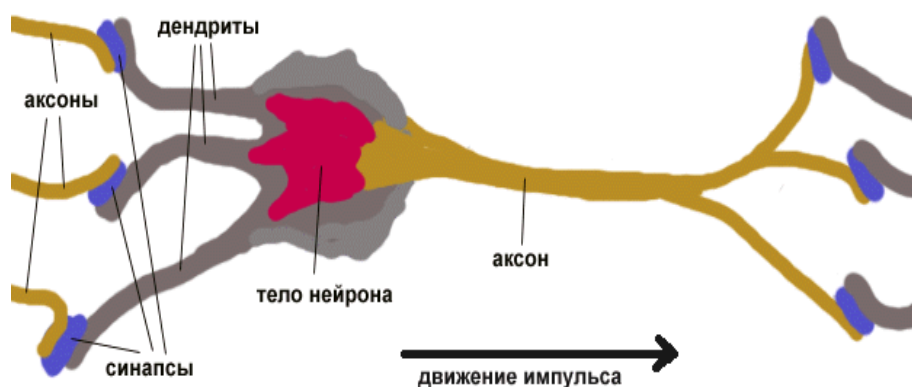


Рис. 9.1. Строение биологического нейрона

Можно считать, что при прохождении синапса сила импульса меняется в определенное число раз, которое мы будем называть весом синапса. Импульсы, поступившие к нейрону одновременно по нескольким дендритам, суммируются. Если суммарный импульс превышает некоторый порог, нейрон возбуждается, формирует собственный импульс и передает его далее по аксону. Важно от-

метить, что веса синапсов могут изменяться со временем, а значит, меняется и поведение соответствующего нейрона. Нетрудно построить математическую модель описанного процесса.

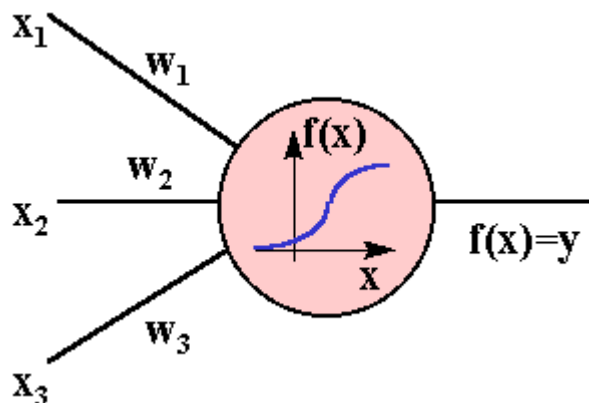


Рис.9.2. Модель персептронного нейрона

На рисунке 9.2 изображена модель персептронного нейрона с тремя входами (дендритами), причем синапсы этих дендритов имеют веса w_1 , w_2 , w_3 . Пусть к синапсам поступают импульсы силы x_1 , x_2 , x_3 соответственно, тогда после прохождения синапсов и дендритов к нейрону поступают импульсы w_1x_1 , w_2x_2 , w_3x_3 . Нейрон преобразует полученный суммарный импульс $x=w_1x_1+w_2x_2+w_3x_3$ в соответствии с некоторой передаточной функцией $f(x)$. Сила выходного импульса равна

$$y = f(x) = f(w_1x_1+w_2x_2+w_3x_3).$$

Таким образом, нейрон полностью описывается своими весами w_k и передаточной функцией $f(x)$. Получив набор чисел (вектор) x_k в качестве входов, нейрон выдает некоторое число y на выходе.

Примечание. К весам нейрона может добавляться смещение b , тогда выход $y = f(x) = f(w_1x_1+w_2x_2+w_3x_3+b)$, и нейрон полностью описывается своими весами w_k , смещением b и передаточной функцией $f(x)$.

1.2. Искусственная нейронная сеть

Искусственная нейронная сеть (ИНС), нейронная сеть - это набор нейронов, соединенных между собой. Как правило, передаточные функции всех нейронов в нейронной сети фиксированы, а веса являются параметрами нейронной сети и могут изменяться. Некоторые входы нейронов помечены как внешние входы нейронной сети, а некоторые выходы - как внешние выходы нейронной сети. Подавая любые числа на входы нейронной сети, мы получаем какой-то набор чисел на выходах нейронной сети. Таким образом, работа нейронной сети состоит в преобразовании входного вектора в выходной век-

тор, причем это преобразование задается весами нейронной сети. Чем больше нейронов в сети, тем более сложное преобразование она выполняет. Практически любую задачу можно свести к задаче, решаемой нейронной сетью. Процесс построения ИНС включает два этапа:

- 1) выбор типа (архитектуры) нейронной сети;
- 2) подбор весов (обучение) нейронной сети.

Типы ИНС. Наиболее часто используются персептронные нейроны с пороговой, сигмоидальной логистической, сигмоидальной тангенциальной и линейной передаточными функциями, а также нейроны с радиальными базисными передаточными функциями. По архитектуре связей ИНС могут быть сгруппированы в два класса: сети прямого распространения и рекуррентные сети или сети с обратными связями. В наиболее распространённом семействе сетей первого класса, называемых многослойным персептроном, нейроны расположены слоями и имеют однонаправленные связи между слоями. Все нейроны одного слоя имеют одинаковые передаточные функции. Пример двухслойного персептрона приведён на рис.9.3.

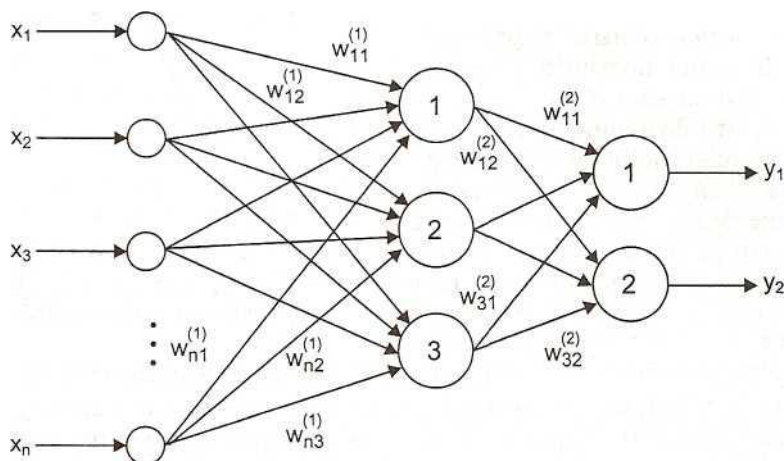


Рис.9.3 Двухслойный персептрон:

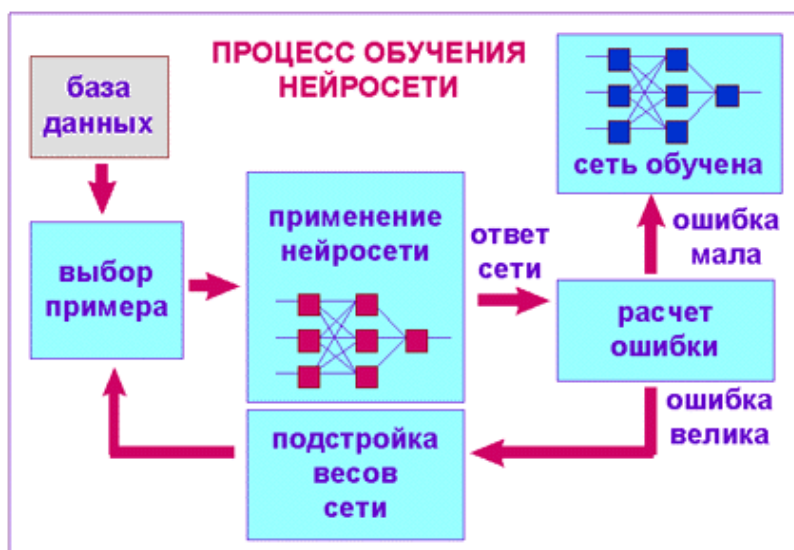
$x_1, x_2, x_3, \dots, x_n$ – входы ИНС, $W_{ij}^{(1)}$ – веса нейронов скрытого слоя, $W_{ij}^{(2)}$ – веса нейронов выходного слоя, y_1, y_2 – выходы ИНС

Число входов ИНС, а также число выходов и, следовательно, число нейронов в выходном слое определяются условиями решаемой задачи. Число скрытых слоёв, число нейронов в каждом скрытом слое и вид передаточной функции определяются методом проб и ошибок для каждой конкретной задачи. Существующие теоретические результаты дают лишь слабые ориентиры для выбора этих параметров в практических приложениях. Чем сложнее связь между входными и выходными переменными, тем большее число нейронов в скрытом слое необходимо предусмотреть. При выборе числа скрытых слоёв приходится искать компромиссное решение, т.к. чем больше скрытых слоёв, тем более высокая точность обучения, но тем ниже обобщающая способность. Для решения большинства практических задач достаточно всего одного промежу-

точного слоя, два слоя применяются как резерв в особых случаях, а сети с тремя слоями практически не применяются. Выбранная архитектура определяет число весов, которые будут подстраиваться в процессе обучения.

ПРИМЕЧАНИЕ. В литературе нет единого мнения относительно того, что именно считать числом слоев в таких сетях. Одни авторы используют число слоев нейронов (включая несуммирующий входной слой), другие — число слоев весов. Так как последнее определение - функционально описательное, то оно будет использовано и нами.

Обучение ИНС. "Обучить" выбранную нейронную сеть означает подобрать такие значения ее весов, чтобы она работала нужным образом. Для конструирования процесса обучения необходимо определить *парадигму, правило и алгоритм обучения*. Существуют три парадигмы обучения: «с учителем», «без учителя» и смешанная. В первом случае ИНС располагает правильными ответами (выходами сети) на каждый входной пример – это наиболее часто используемый в практических задачах вариант. Во втором случае не требуется знание правильных ответов, сеть обучается исходя из внутренней структуры данных, что используется, как правило, при решении задач кластеризации. Известны 4 основные правила обучения ИНС: коррекция по ошибке, машина Больцмана, правило Хебба и обучение методом соревнования. Рассмотрим наиболее часто используемое правило – **коррекция по ошибке** (при обучении с учителем). Имеется некоторая база данных, содержащая примеры (входы и выходы реального объекта). Предъявляя на вход нейронной сети значение входа из примера, мы получаем от нее некоторый ответ, не обязательно верный. Нам известен и верный (желаемый) ответ - значение выхода из примера. Вычисляя разность между желаемым ответом и реальным ответом сети, мы получаем *вектор ошибки*, который используется для модификации весов и смещений, обеспечивающей постепенное уменьшение ошибки.



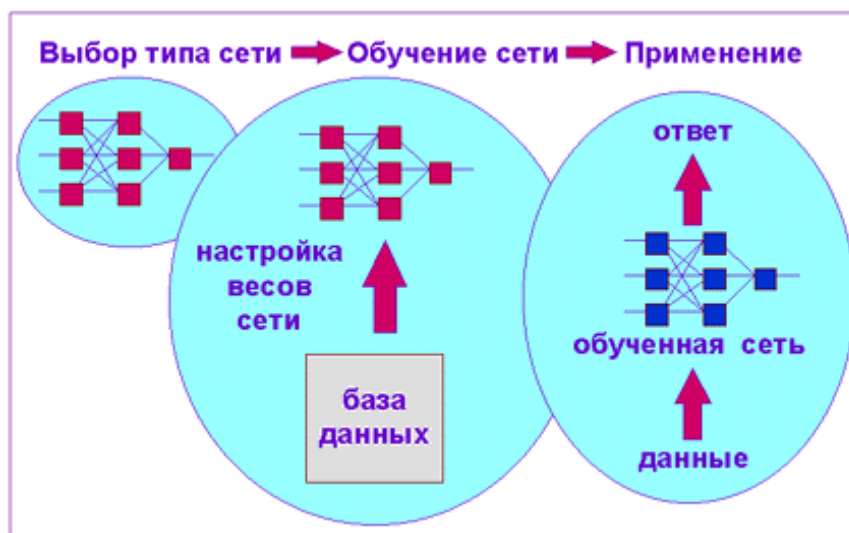
Таким образом, при обучении решается задача оптимизации, в которой оптимизируемыми переменными являются веса и смещения, а целевой функцией ошибка, которую необходимо минимизировать. Для решения этой задачи могут быть использованы следующие (итерационные) алгоритмы:

- 1) алгоритмы локальной оптимизации с вычислением частных производных первого порядка:
 - градиентный алгоритм (метод скорейшего спуска),
 - методы с одномерной и двумерной оптимизацией целевой функции в направлении антиградиента,
 - метод сопряженных градиентов,
 - методы, учитывающие направление антиградиента на нескольких шагах алгоритма;
- 2) алгоритмы локальной оптимизации с вычислением частных производных первого и второго порядка:
 - метод Ньютона,
 - методы оптимизации с разреженными матрицами Гессе,
 - квазиньютоновские методы,
 - метод Гаусса-Ньютона,
 - метод Левенберга-Марквардта;
- 3) стохастические алгоритмы оптимизации:
 - поиск в случайном направлении,
 - имитация отжига,
 - метод Монте-Карло (численный метод статистических испытаний);
- 4) алгоритмы глобальной оптимизации (задачи глобальной оптимизации решаются с помощью перебора значений переменных, от которых зависит целевая функция).

В многослойных сетях оптимальные выходные значения нейронов всех слоёв, кроме последнего, неизвестны, и трёх- или более слойный персептрон уже невозможно обучить, руководствуясь только величинами ошибок на выходах НС. Наиболее приемлемым вариантом обучения оказался градиентный метод поиска минимума функции ошибки с рассмотрением сигнала ошибки от выходов НС к её входам, т.е. в направлении, обратном прямому распространению сигналов в обычном режиме работы. Этот алгоритм обучения НС получил название **алгоритма обратного распространения ошибки** (Error Back Propagation). Функция ошибки представляет собой сумму квадратов отклонений эталонных и расчётных выходов сети. Алгоритм действует циклически, и его циклы принято называть эпохами. На каждой эпохе на вход сети подаются обучающие наблюдения, выходные значения сети сравниваются с эталонными и вычисляется ошибка. Значение ошибки, а также градиента поверхности ошибок используются для корректировки весов, после чего все действия повторяются. Процесс обучения прекращается либо когда пройдено определённое количество эпох, либо когда ошибка достигнет заданного уровня малости, либо когда ошибка перестанет уменьшаться.

Оказывается, что после многократного предъявления примеров веса нейронной сети стабилизируются, причем нейронная сеть дает правильные от-

веты на все (или почти все) примеры из базы данных. В таком случае говорят, что "нейронная сеть выучила все примеры", "нейронная сеть обучена", или "нейронная сеть натренирована". Важно отметить, что вся информация, которую нейронная сеть имеет о задаче, содержится в наборе примеров. Поэтому качество обучения нейронной сети напрямую зависит от количества примеров, на которых обучалась сеть, а также от того, насколько полно эти примеры описывают данную задачу. Так, например, бессмысленно использовать нейронную сеть для предсказания финансового кризиса, если в базе данных кризисов не представлено. Считается, что для полноценной тренировки нейронной сети требуется хотя бы несколько десятков (а лучше сотен) примеров. Для того, чтобы сеть работала правильно не только на тех примерах, на которых обучалась, но и на всех других, поступают следующим образом. Вся совокупность имеющихся примеров делят на 3 множества: обучающее, контрольное и тестовое. На обучающей выборке сеть тренируют, на контрольной выборке проверяют и подстраивают, а на тестовой сеть проходит окончательное испытание на пригодность для использования.



После того, как нейронная сеть обучена и прошла испытания на контрольном и тестовом множествах, мы можем применять ее для решения поставленной задачи – с большой вероятностью ИНС будет правильно реагировать на новые, не предъявленные ей ранее данные.

1.2. Области применения нейронных сетей

Нейросетевая классификация. Отметим, что задачи классификации (типа распознавания букв) очень плохо алгоритмизируются. Любая НС, имитирующая процесс распознавания, работает в двух режимах: в режиме обучения и в режиме распознавания. В режиме обучения некто (человек, машина, робот или природа), играющий роль учителя, предъявляет машине объекты и о каждом из них сообщает, к какому понятию (классу) он принадлежит. По этим данным определяются синаптические веса. В режиме распознавания машине

предъявляются новые объекты (вообще говоря, отличные от ранее предъявленных), и она должна их классифицировать, по возможности, правильно. Примером такой задачи служит медицинская диагностика, где нейронная сеть может учитывать большое количество числовых параметров (энцефалограмма, давление, вес и т.д.) и в зависимости от их значений относить пациента к той или иной группе. Конечно, "мнение" нейронной сети в этом случае нельзя считать окончательным. Классификация предприятий по степени их перспективности - это уже привычный способ использования нейронных сетей в практике западных компаний. При этом нейронная сеть также использует множество экономических показателей, сложным образом связанных между собой.

Для решения задач классификации часто используют многослойный персептрон, вероятностную нейронную сеть, линейные сети.

Нейросетевая кластеризация. Помимо задач классификации, нейронные сети широко используются для решения задач кластеризации. Кластеризация - это разбиение набора примеров на несколько компактных областей (кластеров), причем число кластеров заранее неизвестно. Для решения задач кластеризации используют сети Кохонена и др.

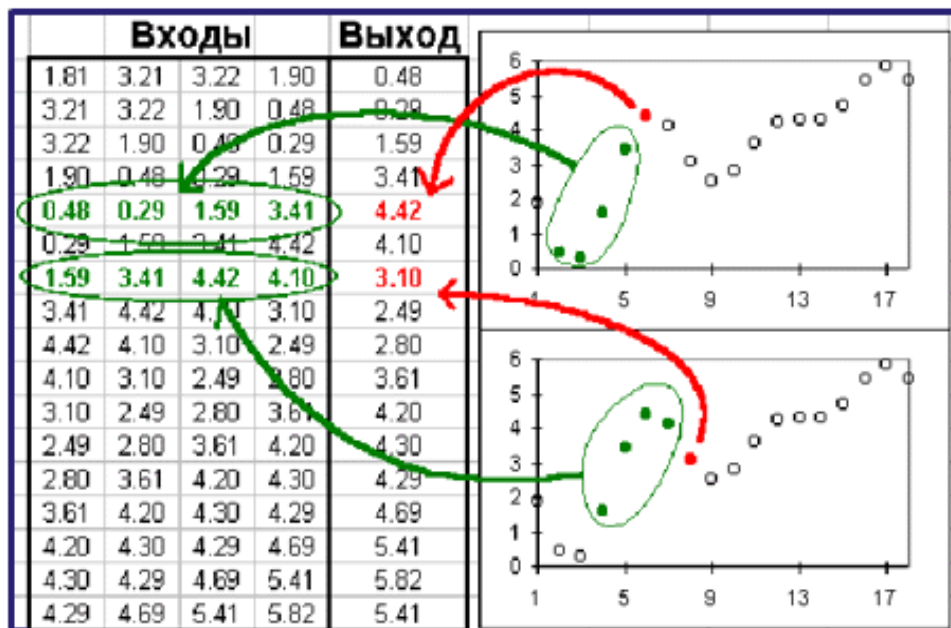
Аппроксимация функций. Предположим, что имеется обучающая выборка (пары данных вход x_i -выход y_i), которая генерируется неизвестной функцией $F(x)$, искаженной шумом. Задача аппроксимации состоит в нахождении оценки неизвестной функции $F(x)$. Функция будет представлена обученной НС, которая продуцирует значения y для любых значений x , не входящих в обучающую выборку. Аппроксимация функций необходима при решении многочисленных инженерных и научных задач моделирования. Для решения таких задач используются обобщенно-регрессионные НС, сети с радиальными базисными элементами, линейные и др.

Нейросетевое прогнозирование. Задачи прогнозирования особенно важны для практики, в частности, для финансовых приложений, поэтому поясним способы применения нейронных сетей в этой области более подробно.

Рассмотрим практическую задачу, ответ в которой неочевиден - задачу прогнозирования курса акций на 1 день вперед. Пусть у нас имеется база данных, содержащая значения курса за последние 300 дней. Простейший вариант в данном случае - попытаться построить прогноз завтрашней цены на основе курсов за последние несколько дней. Понятно, что прогнозирующая нейронная сеть должна иметь всего один выход и столько входов, сколько предыдущих значений мы хотим использовать для прогноза - например, 4 последних значения. Составить обучающий пример очень просто - входными значениями нейронной сети будут курсы за 4 последовательных дня, а желаемым выходом нейронной сети - известный нам курс в следующий день за этими четырьмя.

Если нейронная сеть совместима с какой-либо системой обработки электронных таблиц (например, Excel), то подготовка обучающей выборки состоит из следующих операций:

- скопировать столбец данных значений котировок в 4 соседних столбца;
- сдвинуть второй столбец на 1 ячейку вверх, третий столбец - на 2 ячейки вверх и т.д.



Смысл этой подготовки легко увидеть на рисунке - теперь каждая строка таблицы представляет собой обучающий пример, где первые 4 числа - входные значения нейронной сети, а пятое число - желаемое значение выхода нейронной сети.

Как видно из этого примера, объем обучающей выборки зависит от выбранного нами количества входов нейронной сети. Если сделать 299 входов, то такая нейронная сеть потенциально могла бы строить лучший прогноз, чем нейронная сеть с 4 входами, однако в этом случае мы имеем всего 1 обучающий пример, и обучение бессмысленно. При выборе числа входов нейронной сети следует учитывать это, выбирая разумный компромисс между глубиной предсказания (число входов нейронной сети) и качеством обучения нейронной сети (объем тренировочного набора).

2. ПРАКТИЧЕСКАЯ РЕАЛИЗАЦИЯ МЕТОДОЛОГИИ НЕЙРОННЫХ СЕТЕЙ

Рассмотрим практическую реализацию нейросетевых алгоритмов в MATLAB на примерах решения конкретных задач из различных областей применения НС.

ЗАДАНИЕ №1

1. Построить графики передаточных функций (функций активации):

- пороговые функции активации с жестким ограничением **hardlim**, **hardlims**;
- линейная функция активации **purelin**;
- сигмоидальная логистическая функция активации **logsig**;
- сигмоидальная функция гиперболический тангенс **tansig**

(Примеры в приложении)

2. На конкретном примере освоить основные этапы реализации нейросетевого подхода.

Выделим 5 основных этапов:

1. Подготовка данных для обучения сети.
2. Создание сети.
3. Обучение сети.
4. Тестирование сети.
5. Моделирование сети. (Использование сети для решения поставленной задачи.)

В качестве примера рассмотрим следующую задачу. Пусть задан массив, состоящий из нескольких значений функции

$$y = c \cdot \exp(-(x - a)^2 / S); \quad (S > 0)$$

на интервале изменения x (0,1). Создать нейронную сеть такую, что при вводе этих значений Y на **входы** сети на **выходах** получались бы значения параметров c , a и s .

1. Подготовка данных для обучения сети

В первую очередь необходимо определиться с размерностью входного массива. Выберем количество значений функции равным $N = 21$, т.е. в качестве входных векторов массива используем значения функции y в точках $x = 0; 0.05; \dots 1.0$. Для обучения сети необходимо сформировать массив входных векторов для различных наборов параметров c , a и s . Каждый набор этих параметров является **вектором-эталонном** для соответствующего **входного вектора**.

Для подготовки входного и эталонного массивов воспользуемся следующим алгоритмом. Выбираем случайным образом значения компонент вектора – эталона C , A , S и вычисляем компоненты соответствующего входного вектора. Повторяем эту процедуру M раз и получаем массив входных векторов в виде матрицы размерностью $N \times M$ и массив векторов – эталонов в виде матрицы размерностью в нашем случае $3 \times M$. Полученные массивы мы можем использовать для обучения сети. Прежде чем приступить к формированию обучающих массивов, необходимо определиться с некоторыми свойствами массивов.

Диапазон изменения параметров C , A , S . Выберем диапазоны изменения параметров C , A , S равными (0.1, 1). Значения, близкие к 0 и сам 0 исключим в связи с тем, что функция не определена при $S = 0$. Второе ограничение связано с тем, что при использовании типичных передаточных функций желательно, чтобы компоненты входных и выходных векторов не выходили за пределы диапазона (-1,1). В противном случае можно использовать методы нормировки, которые позволяют обойти это ограничение.

Количество входных и эталонных векторов выберем равным $M = 100$. Этого достаточно для обучения, а процесс обучения не займет много времени.

Обучающий, тестовый массивы и эталоны подготовим с помощью программы *mas1*:

```
% формирование массивов (входной массив P и эталоны T)
P=zeros(100,21);
```

```

T=zeros(3,100);
x=0:5.e-2:1;
for i=1:100
c=0.9*rand+0.1;
a=0.9*rand+0.1;
s=0.9*rand+0.1;
T(1,i)=c;
T(2,i)=a;
T(3,i)=s;
P(i,:)=c*exp(-((x-a).^2/s));
end;
P=P';

```

С помощью этой программы формируется матрица P из $M = 100$ входных векторов-столбцов, каждый из которых сформирован из 21 точки исходной функции со случайно выбранными значениями параметров C , A , S , и матрица T эталонов из 100 эталонных векторов-столбцов, каждый из которых сформирован из 3 соответствующих эталонных значений. Матрицы P и T будут использованы при обучении сети. Следует отметить, что при каждом новом запуске этой программы будут формироваться массивы с новыми значениями компонентов векторов, как входных, так и эталонных.

2. Создание сети

Вообще, выбор архитектуры сети для решения конкретной задачи основывается на опыте разработчика. Поэтому предложенная ниже архитектура сети является одним вариантом из множества возможных конфигураций. Сформируем однонаправленную сеть с числом входов 21 (по числу элементов входного вектора), включающую два скрытых слоя. В первом слое 10 нейронов с передаточной функцией `logsig`, во втором слое 15 нейронов с передаточной функцией `logsig` и в выходном слое 3 нейрона (по числу компонентов выходного вектора) с передаточной функцией `purelin`. При этом в качестве обучающего алгоритма выберем алгоритм Levenberg-Marquardt (`trainlm`). (Функции создания и тренировки сети, а также функции активации, доступные в Neural Network Toolbox, приведены в приложении). По умолчанию используется `trainlm`. Указанная сеть формируется с помощью процедуры:

```

net=
newff(minmax(P),[10,15,3],{'logsig','logsig','purelin'},'trainlm');

```

Первый аргумент - матрица 21×2 минимальных и максимальных значений сигнала на каждом из входов - вычисляется с помощью процедуры `minmax`. Результатом выполнения процедуры `newff` является объект - нейронная сеть `net` заданной конфигурации. Сеть можно сохранить на диске в виде `mat`. файла с помощью команды `save` и загрузить снова с помощью команды `load`. Более подробную информацию о процедуре можно получить, воспользовавшись командой `help`.

3. Обучение сети

Следующий шаг – обучение созданной сети. Перед обучением необходимо задать параметры обучения. Задаем функцию оценки функционирования sse.

```
net.performFcn='sse';
```

В этом случае в качестве оценки вычисляется сумма квадратичных отклонений выходов сети от эталонов. Задаем критерий окончания обучения – значение отклонения, при котором обучение будет считаться законченным:

```
net.trainParam.goal=0.01;
```

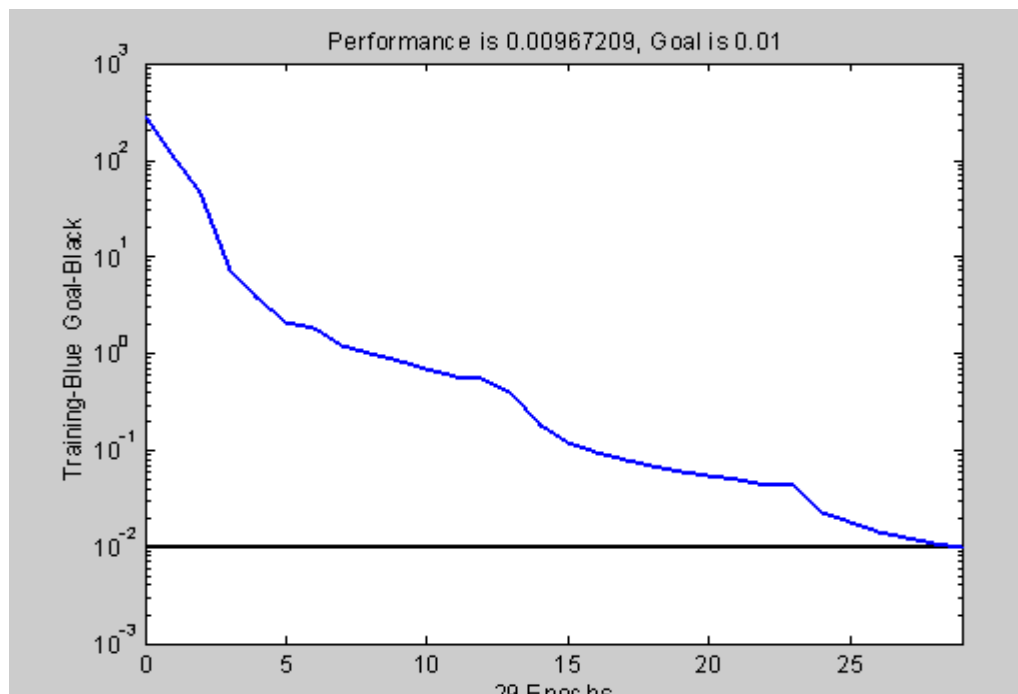
Задаем максимальное количество циклов обучения. После того, как будет выполнено это количество циклов, обучение будет завершено:

```
net.trainParam.epochs=1000;
```

Теперь можно начинать обучение:

```
[net,tr]=train(net,P,T);
```

Процесс обучения иллюстрируется графиком зависимости оценки функционирования от номера цикла обучения (номера эпохи).



Таким образом, обучение сети окончено. Теперь эту сеть можно сохранить в файле nn1.mat:

```
save nn1 net;
```

В последующих сеансах работы сохранённую сеть можно загрузить, используя функцию

```
load('nn1').
```

4. Тестирование сети

Перед тем, как воспользоваться нейронной сетью, необходимо исследовать степень достоверности результатов вычислений сети на тестовом массиве входных векторов. В качестве тестового массива необходимо использовать массив, компоненты которого отличаются от компонентов массива, использованного для обучения. В нашем случае для получения тестового массива достаточно воспользоваться еще раз программой `mas1`.

Для оценки достоверности результатов работы сети можно воспользоваться результатами регрессионного анализа, полученными **при сравнении эталонных значений со значениями, полученными на выходе сети, когда на вход поданы входные векторы тестового массива**. В среде MATLAB для этого можно воспользоваться функцией `postreg`. Следующий набор команд иллюстрирует описанную процедуру:

```
>> mas1 - создание тестового массива P
>> y=sim(net,P); - обработка тестового массива обученной сетью, y - выходы сети.
>> [m,b,r]=postreg(y(1,:),T(1,:));
```

При выполнении последней команды выводится график в координатах: по оси ординат A - значения, полученные на первом выходе сети (расчетные значения коэффициента C), по оси абсцисс T – эталонные значения коэффициента c . Совершенно очевидно, что если все точки группируются вблизи прямой

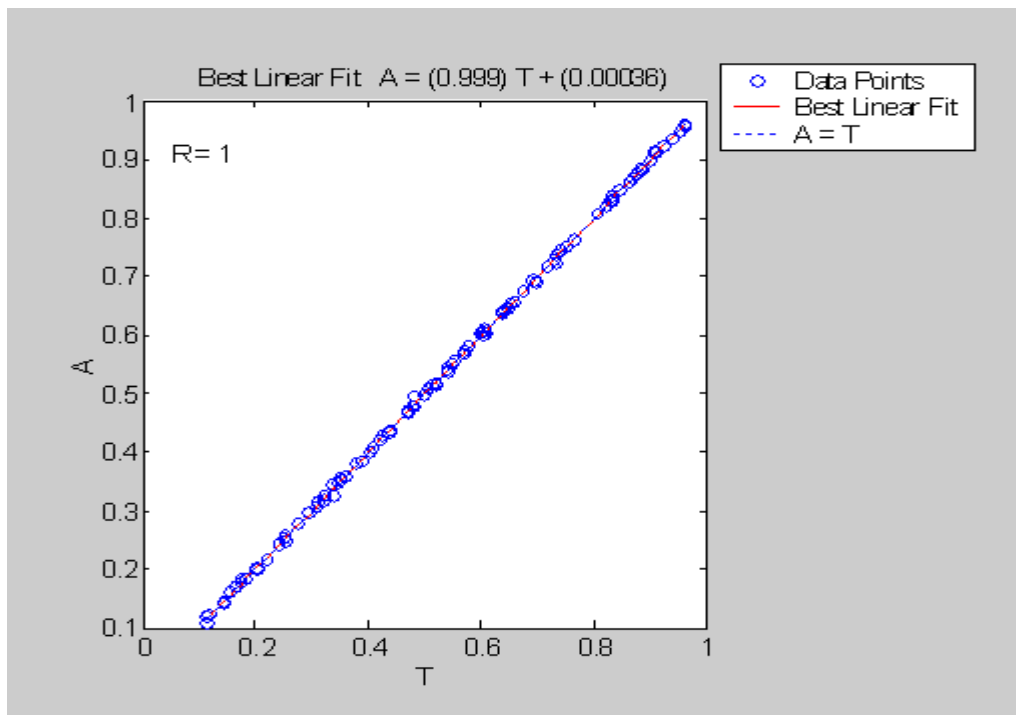
$$A = T, \quad (1)$$

то сеть на тестовом массиве работает правильно. Над графиком выводится уравнение прямой, найденное по полученным данным в виде:

$$A = m \cdot T + b. \quad (2)$$

Зависимость (2) тем точнее представляет (1), чем значение m ближе к 1, а значение b к 0.

Кроме того, выводится коэффициент корреляции R (в параметрах команды обозначен r).



Аналогичным образом проверяем достоверность результатов работы сети для оставшихся коэффициентов.

```
>> [m,b,r]=postreg(y(2,:),T(2,:));- соответствие расчётных и эталонных значений коэффициента a.
```

```
>> [m,b,r]=postreg(y(3,:),T(3,:));- соответствие расчётных и эталонных значений коэффициента s.
```

Графики показывают, что сеть хорошо решает поставленную задачу для всех трех выходных параметров и можно перейти к её использованию.

Сохраним обученную сеть `net` на диске в файл `nn1.mat`

```
save nn1 net
```

5. Моделирование сети. (Использование сети для решения поставленной задачи)

Для того чтобы применить обученную сеть для обработки данных, необходимо воспользоваться функцией `sim`:

```
Y=sim(net,p);
```

где `p` – набор входных векторов, `Y` – результат анализа в виде набора выходных векторов.

В качестве входного вектора можно выбрать один из векторов, например:

```
p(:,25):
```

```
Y=sim(net,p(:,25));
```


Полученное значение Y должно быть близким к соответствующему эталону:

$T(:, 25)$

Можно самим задать любые значения коэффициентов, рассчитать значения входного вектора, определить коэффициенты на выходе сети и сравнить их с заданными.

Например, пусть

$c=0.2, a=0.8, s=0.7$, тогда значения на входе сети

$p=0.2 \cdot \exp(-((x-0.8).^2/0.7));$

$p=p'$

Подставив этот входной вектор в качестве аргумента функции `sim`:

$Y=\text{sim}(\text{net}, p),$

получим:

$Y =$
 0.2048 (C)
 0.8150 (A)
 0.7048 (S)

Близость полученных значений к правильному результату (0.2; 0.8; 0.7) свидетельствует о применимости сети. Если же результаты сильно отличаются, необходимо изменить архитектуру сети, точность или метод поиска.

Задания для самостоятельной работы

Вариант	Функция	Вариант	Функция
1	$Y = \frac{a}{b + x^c}$	9	$Y = \frac{b + ax^2}{c}$
2	$Y = a + \frac{b}{x^2 + c}$	10	$Y = a + b \cdot x + c \cdot x^2$
3	$Y = a + b \cdot \ln x$	11	$Y = \exp(a + b \cdot x + c \cdot x^2)$
4	$Y = a \cdot x^b$	12	$Y = c + \frac{a \cdot \ln x}{b \cdot x}$
5	$Y = a \cdot \exp(b \cdot x) + c$	13	$Y = \frac{a + b \cdot x^2}{c}$

Вариант	Функция	Вариант	Функция
6	$Y = \frac{b}{a \cdot \exp(-x)}$	14	$Y = a \cdot \exp(\frac{b \cdot x^2}{c})$
7	$Y = c \cdot \exp(\frac{-2 \cdot x}{a + b \cdot x})$	15	$Y = \frac{a \cdot x^b}{c}$
8	$Y = (a \cdot x + b)^3 / c$	16	$Y = a \cdot \exp(b \cdot x^2)$

Диапазон изменения x и параметров a , b , c выбрать произвольно, исходя из вида функции.

2.1. Аппроксимация функций с использованием НС

ЗАДАНИЕ №2. Создать нейронную сеть, реализующую функциональную зависимость $Y = F(x)$ между входом (x) и выходом (Y). Вид зависимости взять из задания №1. Значения коэффициентов в этом случае заданы (можно взять любую комбинацию значений из решения предыдущей задачи). Последовательность решения должна включать все 5 этапов, приведённые в задании №1. О выборе типа сети для аппроксимации функций - в разделе 1.3.

2.2. Формирование нейросетевых моделей в Simulink

ЗАДАНИЕ №3. Основной функцией для формирования нейросетевых моделей в Simulink является функция

```
gensim(net, st),
```

где **net** – имя созданной нейронной сети (НС), **st** – интервал дискретизации (если НС не имеет задержек, ассоциированных с её входами или слоями, **st** = -1).

В качестве примера рассмотрим следующую задачу.

Пусть входной и целевой (эталонный) вектор имеют вид $P = [1 \ 2 \ 3 \ 4 \ 5]$; $t = [1 \ 3 \ 5 \ 7 \ 9]$.

Создадим однослойную линейную НС с одним входом, одним выходом и одним нейроном:

```
>> p=[1 2 3 4 5];
>> t=[1 3 5 7 9];
>> net=newlind(p,t);
```

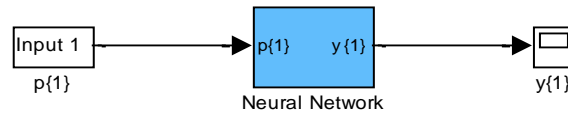
Протестируем ее:

```
>> y=sim(net,p)
y =
    1.0000    3.0000    5.0000    7.0000    9.0000
```

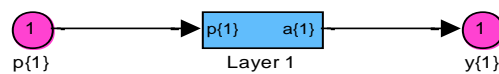
Затем запустим Simulink командой

```
>> gensim(net,-1)
```

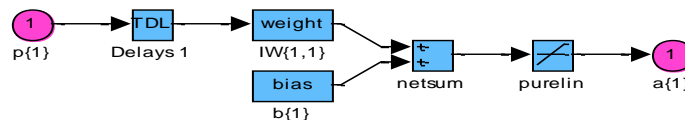
Появится окно S-модели созданной НС (Neural Network)



Открыв подсистему Neural Network, можно увидеть структуру сети (Layer-слой):



а подсистема Layer 1 представляет собой структуру нейрона:



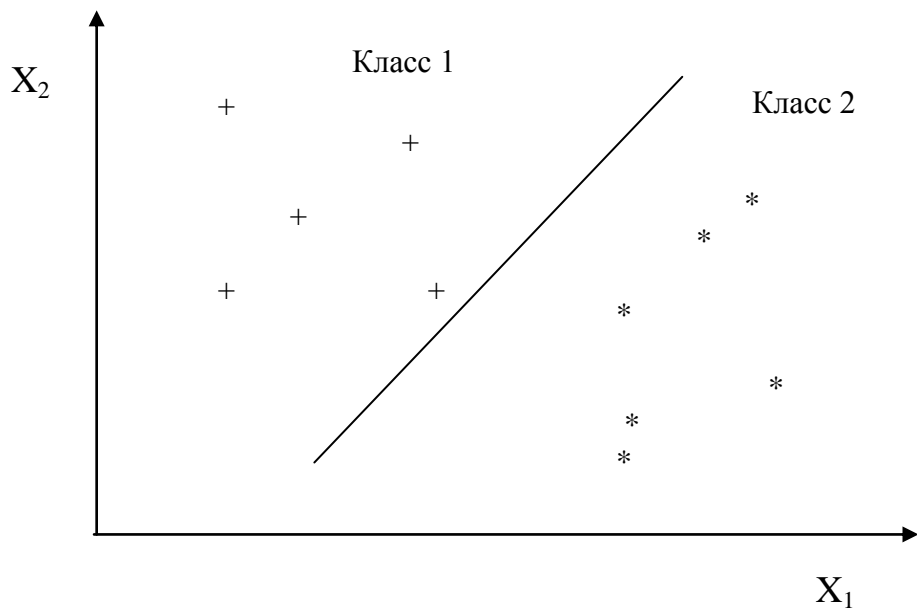
Пакет Neural Network Toolbox содержит ряд блоков, которые могут быть использованы для построения нейронных сетей в среде Simulink, либо применяться вместе с рассмотренной выше функцией gensim.

Для вызова этого набора в командной строке необходимо набрать команду **neural**. Каждый из представленных в окне блоков, в свою очередь, является соответствующей библиотекой.

Задание для самостоятельной работы. Сформируйте модель НС, рассмотренную в задании №2 в Simulink, используя функцию gensim. Обратите внимание на структуру НС в S – модели.

2.3. Классификация с помощью ИНС

ЗАДАНИЕ №3. Выполнить генерацию исходных данных в виде точек на плоскости, разделённой прямой линией $X_2 = a + b \cdot X_1$. Точки, попавшие по одну сторону линии, будут относиться к классу 1, по другую – к классу 2. Создать нейронную сеть, способную отнести любую новую точку к соответствующему классу.



Значения коэффициентов в уравнении прямой

Номер варианта	a	b	Номер варианта	a	b
1	0.4	1.5	9	0.8	0.3
2	-0.2	2.1	10	-0.8	0.4
3	0.5	1.6	11	0.2	1.7
4	0.7	0.9	12	-0.3	0.2
5	-0.4	1.8	13	-0.6	0.5
6	0.3	1.9	14	0.9	0.6
7	-0.7	1.2	15	1.2	1.1
8	0.6	0.7	16	-1.1	1.3

2.4. Прогнозирование значений процесса, заданного функцией времени

ЗАДАНИЕ №4. Рассмотрим пример прогнозирования с помощью простейшей линейной НС процесса, поведение которого зависит от времени. В этом примере надо особое внимание уделить способу формирования входов и выходов НС.

Предположим, что имеется сигнал (функция времени), описываемый соотношением

$$x(t) = \sin(4\pi \cdot t),$$

который подвергается дискретизации с интервалом 0.025 с.

Построим линейную НС, позволяющую прогнозировать будущее значение по 5 предыдущим.

```
t=0:0.025:5; % Задание диапазона времени от 0 до 5 секунд
x=sin(t*4*pi); % предсказываемый сигнал
Q=length(x); %число столбцов в матрице входов
% Создание входных векторов
P=zeros(5,Q);
P(1,2:Q)=x(1,1:(Q-1));
P(2,3:Q)=x(1,1:(Q-2));
P(3,4:Q)=x(1,1:(Q-3));
P(4,5:Q)=x(1,1:(Q-4));
P(5,6:Q)=x(1,1:(Q-5));
s=newlind(P,x); % Создание новой НС с именем s
y=sim(s,P); % расчёт прогнозируемых значений
% Создание графиков исходного сигнала и прогноза
plot(t,x,t,y,'*')
xlabel('Vremja')
ylabel('Прогноз - сигнал *');
title('Выход сети - прогноз');
%Ошибка прогноза
e=y-x;
% последние 5 значений, рассчитанные по функции и по нейронной се-
ти
znach=x(Q-5:Q)
prognoz=y(Q-5:Q)
```

3. ВОПРОСЫ ДЛЯ САМОКОНТРОЛЯ ЗНАНИЙ

1. Какого типа задачи решаются с использованием ИНС?
2. Опишите работу биологического нейрона.
3. Как функционирует искусственный нейрон?
4. Какие функции активации используются в искусственных нейронах?
5. Что такое ИНС?
6. Как определяется число входов и выходов ИНС?
7. Перечислите известные Вам типы нейронных сетей.
8. Каким образом функционирует сеть прямого распространения?
9. Каким образом функционирует сеть встречного распространения?
10. В чём заключается суть обучения НС?
11. Когда заканчивается процесс обучения?
12. Опишите алгоритм обучения НС методом обратного распространения ошибки.
13. По какому принципу разделяется массив имеющихся экспериментальных данных?
14. Каким образом происходит классификация с помощью НС?
15. Каким образом происходит кластеризация с помощью НС?

16. Каким образом происходит прогнозирование с помощью НС?

4. ПРИЛОЖЕНИЕ

Функции создания новой сети

- **network** – создание нейронной сети пользователя;
- **newc** – создание конкурентного слоя;
- **newcf** – создание каскадной направленной сети;
- **newelm** – создание сети обратного распространения Элмана (Elman);
- **newff** – создание однонаправленной сети;
- **newfftd** – создание однонаправленной сети с входными задержками;
- **newgrnn** – создание обобщенной регрессионной нейронной сети;
- **newhop** – создание рекуррентной сети Хопфилда;
- **newlin** – создание линейного слоя;
- **newlind** – конструирование линейного слоя;
- **newlvq** – создание квантованной сети;
- **newp** – создание персептрона;
- **newpnn** – конструирование вероятностной нейронной сети;
- **newrb** – конструирование сети с радиальным базисом;
- **newrbe** – конструирование точной сети с радиальными базисными функциями;
- **newsom** – создание самоорганизующейся карты.

Функции активации

- **compet** – конкурирующая функция активации;
- **hardlim** – ступенчатая функция активации
эта функция описывается соотношением
$$a = \text{hardlim}(n) = 1(n)$$

и равна 0, если $n < 0$,
и равна 1, если $n \geq 0$.
Построим график этой функции в диапазоне значений входа от -5 до + 5:
n = -5:0.1:5;
plot(n,hardlim(n),'b+:');
- **hardlims** – ступенчатая симметричная функция активации;
- **logsig** – сигмоидная (логистическая) функция активации
эта функция описывается соотношением
$$a = \text{logsig}(n) = 1/(1 + \exp(-n)).$$

Она принадлежит к классу сигмоидальных функций, и ее аргумент может принимать любое значение в диапазоне от $-\infty$ до $+\infty$, а выход изменяется в диапазоне от 0 до 1. Благодаря свойству дифференцируемости, эта функция часто используется в сетях с обучением на основе метода обратного распространения ошибки.
Построим график этой функции в диапазоне значений входа от -5 до + 5:
n=-5:0.1:5;
plot(n,logsig(n),'b+:');
- **poslin** – положительная линейная функция активации
- **purelin** – линейная функция активации
Эта функция описывается соотношением

$a = \text{purelin}(n) = n.$

Построим график этой функции в диапазоне значений входа от -5 до + 5:

n=-5:0.1:5;

plot(n,purelin(n),'b+:');

- **radbas** – радиальная базисная функция активации
- **satlin** – насыщающаяся линейная функция активации
- **satlins** – симметричная насыщающаяся линейная функция активации
- **softmax** – функция активации, уменьшающая диапазон входных значений
- **tansig** – функция активации гиперболический тангенс
- **tribas** – треугольная функция активации

Функции тренировки

- **trainb** – пакетная тренировка с использованием правил обучения для весов и смещений
- **trainbfg** – тренировка сети с использованием квази-Ньютоновского метода BFGS
- **trainbr** – регуляризация Bayesian
- **trainc** – использование приращений циклического порядка
- **traincgb** – метод связанных градиентов Пауэлла-Била (Powell-Beale)
- **traincgf** - метод связанных градиентов Флетчера-Пауэлла (Fletcher-Powell)
- **traincgp** - метод связанных градиентов Полака-Рибера (Polak-Ribiere)
- **traingd** – метод градиентного спуска
- **traingda** – метод градиентного спуска с адаптивным обучением
- **traingdm** - метод градиентного спуска с учетом моментов
- **traingdx** - метод градиентного спуска с учетом моментов и с адаптивным обучением
- **trainlm** – метод Левенберга-Маркара (Levenberg-Marquardt)
- **trainoss** – одноступенчатый метод секущих
- **trainr** – метод случайных приращений
- **trainrp** – алгоритм упругого обратного распространения
- **trains** – метод последовательных приращений
- **trainscg** - метод шкалированных связанных градиентов

Функции использования сети

- **adapt** – разрешает адаптацию сети
- **disp** – отображает свойства нейронной сети
- **display** – отображает имена переменных и свойства сети
- **init** – инициализация нейронной сети
- **sim** – моделирование нейронной сети
- **train** – тренировка нейронной сети

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Советов Б.Я., Яковлев С.А. Моделирование систем: учеб. для вузов – 3-е изд., перераб. и доп. – М.: Высш. шк., 2001. – 343 с.: ил. ISBN 5-06-003860-2
2. Бобков С.П. Моделирование систем: учеб. Пособие / С.П. Бобков, Д.О. Бытев; Иван. гос. хим.-технол. ун-т. – Иваново, 2008. – 156 с. – ISBN 978-5-9616-0286-6
3. Гультияев А.К. MATLAB 5.2. Имитационное моделирование в среде Windows: практ. пособие. – СПб.: КОРОНА принт, 1999. – 288 с. ISBN 5-7931-0053-9
4. Бережная Е.Б., Бережной В.И. Математические методы моделирования экономических систем: учеб. пособие. – М.: Финансы и статистика, 2001. – 368 с.: ил. ISBN 5-279-02291-8
5. MATLAB 6 / В. Дьяконов. - СПб.: Питер, 2001. - 592 с.: ил. - (Сер. "Учеб. курс"). ISBN 5-318-00363-X.
6. Коршунов Ю.М. Математические основы кибернетики: учеб. пособие для вузов. – 3-е изд., перераб. и доп. – М.: Энергоатомиздат, 1987. – 496 с.

Редактор В. Л. Родичева

Подписано в печать 17.03.2010 Усл. п.л. 7,44 Уч. изд. л. 8,26
Формат 60x84 1/16. Тираж 50 экз. Заказ _____

Ивановский государственный химико-технологический университет
153000, г. Иваново, пр-т Ф. Энгельса, 7

Отпечатано на полиграфическом оборудовании кафедры экономики и финансов ИГХТУ