Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Локтионова Оксана Геннадьевна
Должность: проректор по учебной работе
Дата подписания: 08.09.2023 07:48:17
Уникальный программный ключ:
0b817ca911e6668abb13a5d426d39e5f1c11eabbf73e943df4a4851fda56d089

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования
«Юго-Западный государственный университет»
(ЮЗГУ)

Кафедра Биомедицинской инженерии

УТВЕРЖДАЮ
Проректор по учебной работе
_____ О.Г. Локтионова
«02» _____ 2023 г.

Лабораторный практикум по дисциплине «Медицинская кибернетика»

Методические рекомендации по выполнению практических занятий для студентов специальности 30.05.03 «Медицинская кибернетика»

Курск 2023

УДК 007+517.977+681.51

Составители: М.В. Артеменко

Рецензент
доктор технических наук, профессор Чернецкая И. Е.

Лабораторный практикум по дисциплине «Медицинская кибернетика» / Юго-Зап. гос. ун-т; сост. М.В. Артеменко, 2023. − 222 с.: рисунков 58, приложений - 7, библиография – стр.:59, 64, 82, 94, 112, 118, 127, 151, 161, 186, 208, 220.

Лабораторный практикум содержит краткие теоретические сведения, порядок выполнения и содержание отчета к практическим работам по дисциплине «Медицинская кибернетика» и соответствуют требованиям Федерального государственных образовательных стандартов высшего образования по специальности 30.05.03 «Медицинская кибернетика». Рассматриваются разделы: применения средств MathLab для исследования систем автоматического управления (устойчивость, качественные показатели, частотно-амплитудный анализ, переходные процессы, построения критических областей функционирования), применения линейного, геометрического и динамического программирования при решении оптимизационных задач, выделения информативных признаков, построения диагностических правил, нечеткого управления, планирования эксперимента, сети Кохонена, применение дискриминантного анализа в системах управления. Экспериментальная часть работ основывается на применении компьютерных технологий для обработки результатов исследований.

Предназначено для студентов для студентов специальности 30.05.03 «Медицинская кибернетика»

Текст печатается в авторской редакции

Подписано в печать Формат 60x84x 1/16.
Усл.печ.л. . Уч.-изд.л. . Тираж 100 экз. Заказ. Бесплатно.
Юго-Западный государственный университет.
305040, г.Курск, ул. 50 лет Октября,94

СОДЕРЖАНИЕ

Введение 4

1 Использование средств MathLab для анализа систем управления …… 5

2 Реакция систем управления на типовые входные воздействия. Анализ переходных процессов …….. 60

3 Решение оптимизационных задач методами линейного и геометрического программирований ……. 65

4 Применение динамического программирования при оптимизации САУ БТС ……... 83

5 Выделение ритмических составляющих медико-биологических сигналов …… 95

6 Выделение информативных характеристик из медико-биологических сигналов ……. 113

7 Построение решающих диагностических правил ……... 121

8 Планирование эксперимента …….. 129

9 Исследование поведения функций принадлежностей ……. 153

10 Построение нечеткой системы с использованием пакета Fuzzy Logic Toolbox ……... 163

11 Использование нейронной сети при решении классификационных задач состояния объекта управления ……. 187

12 Применение дискриминантного анализа для построения решающих правил в классификационных компонентах систем управления …… 212

Введение

Структура методических указаний по выполнению комплекса лабораторных работ: цель и задачи работы, информационные материалы (включая краткие теоретические сведения), методические указания по выполнению работы, библиография.

Порядок выполнения.

1. Изучить информационные материалы к занятию, включая рекомендованную литературу и лекции.
2. Выполнить лабораторную работу согласно методическим указаниям.
3. Осуществить обработку полученных результатов.
4. Оформить отчет к работе.
5. Подготовиться к отчету – собеседованию с преподавателем по тематике работы (защите полученных результатов).

Содержание отчета.

1. Цель работы. Постановка задач лабораторных исследований.
2. Этапы проведения, полученные результаты, анализ и выводы.
3. Ответы на контрольные вопросы (не менее 3), аннотации содержания информационных источников (100-200 слов по каждому).

1. ИСПОЛЬЗОВАНИЕ СРЕДСТВ MATLAB ДЛЯ АНАЛИЗА СИСТЕМ УПРАВЛЕНИЯ

Цель: Ознакомление со средствами MatLab для анализа систем автоматического управления.

Часть 1. Введение в MatLab.

Рабочая среда MatLab 6.x, MatLab 7 имеет удобный интерфейс для доступа ко многим вспомогательным элементам MatLab.

При запуске MatLab 6.x на экране появляется рабочая среда, изображенная на рисунке 1.

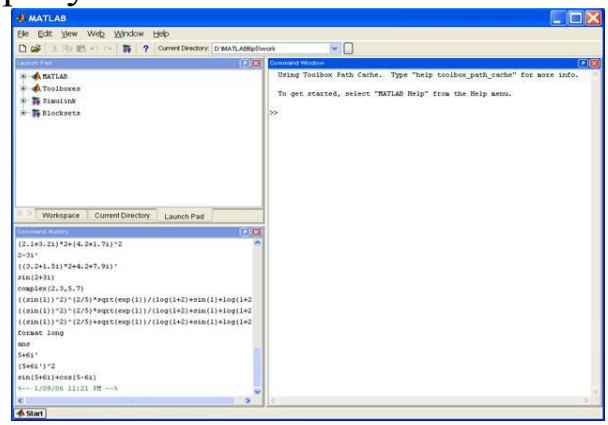


Рис. 1. Рабочая среда пакета MatLab 6.x

Рабочая среда содержит следующие элементы:

- меню;
- панель инструментов с кнопками и раскрывающимся списком;
- окно с вкладками Launch Pad и Workspace, из которого можно получить простой доступ к различным модулям ToolBox и к содержимому рабочей среды;
- окно с вкладками Command History и Current Directory, предназначенное для просмотра и повторного вызова ранее введенных команд, а также для установки текущего каталога;
- командное окно Command Window с командной строкой, в которой находится мигающий курсор;
- строку состояния.

Все команды следует набирать в командной строке. Сам символ », обозначающий приглашение командной строки, приведенный в примерах, набирать не нужно. Для просмотра рабочей области удобно использовать полосы скроллинга или клавиши <Home>, <End> для перемещения влево или вправо и <PageUp>, <PageDown> для перемещения вверх или вниз. Про использование клавиш <up>, <down>, <rigth>, <left> будет сказано дополнительно. Если вдруг после перемещения по рабочей области командного окна пропала командная строка с мигающим курсором, просто нажмите <Enter>.

Важно запомнить, что набор любой команды или выражения должен заканчиваться нажатием на клавишу <Enter> для того, чтобы программа MatLab выполнила эту команду или вычислила выражение.

Если в рабочей среде MatLab 6.x отсутствуют некоторые описанные окна, то следует в меню View выбрать соответствующие пункты: Command Window, Command History, Current Directory, Workspace, Launch Pad.

Арифметические вычисления

Встроенные математические функции MatLab позволяют находить значения различных выражений. MatLab предоставляет возможность управления форматом вывода результата. Команды для вычисления выражений имеют вид, свойственный всем языкам программирования высокого уровня.

Наберите в командной строке 1+2 и нажмите <Enter>. В результате в командном окне MatLab отображается следующее:

```
» 1+2
ans =
3
» |
```

Что сделала программа MatLab? Сначала она вычислила сумму 1+2, затем записала результат в специальную переменную ans и вывела ее значение, равное 3, в командное окно. Ниже ответа расположена командная строка с мигающим курсором, обозначающая, что MatLab готова к дальнейшим вычислениям.

Можно набирать в командной строке новые выражения и находить их значения.

Если требуется продолжить работу с предыдущим выражением, например, вычислить (1+2)/4.5, то проще всего воспользоваться уже имеющимся результатом, который хранится в переменной ans. Наберите в командной строке ans/4.5 (при вводе десятичных дробей используется точка) и нажмите <Enter>, получается:

```
» ans/4.5
ans =
0.6667
» |
```

Вид, в котором выводится результаты вычислений, зависит от формата вывода, установленного в MatLab. Далее объяснено, как задать основные форматы вывода.

Форматы вывода результата вычислений

Требуемый формат вывода результата определяется пользователем из меню MatLab. Выберите в меню File пункт Preferences. На экране появится диалоговое окно Preferences. Для установки формата вывода следует убедиться, что в списке левой панели выбран пункт Command Window. Задание формата производится из раскрывающегося списка Numeric format панели Text display.

Разберем пока только наиболее часто используемые форматы. Выберите short в раскрывающемся списке Numeric format в MatLab 6.x. Закройте диалоговое окно, нажав кнопку ОК. Сейчас установлен короткий формат с плавающей точкой short для вывода результатов вычислений, при котором на экране отображаются только четыре цифры после десятичной точки. Наберите в командной строке 100/3 и нажмите <Enter>. Результат выводится в формате short:

```
» 100/3
ans =
33.3333
```

Этот формат вывода сохранится для всех последующих вычислений, если только не будет установлен другой формат. Заметьте, что в MatLab возможна ситуация, когда при отображении

слишком большого или малого числа результат не укладывается в формат short. Вычислите 100000/3, результат выводится в экспоненциальной форме:

```
» 100000/3
ans =
3.3333e+004
```

То же самое произойдет и при нахождении 1/3000:

```
» 1/3000
ans =
3.3333e-004
```

Однако, первоначальная установка формата сохраняется и при дальнейших вычислениях, для небольших чисел вывод результата снова будет происходить в формате short.

В предыдущем примере пакет MatLab вывел результат вычислений в экспоненциальной форме. Запись 3.3333e-004 обозначает 3.3333*10-4 или 0.00033333. Аналогично можно набирать числа в выражениях. Например, проще набрать 10e9 или l.0e10, чем 1000000000, а результат будет тот же самый. Пробел между цифрами и символом е при вводе не допускается, т.к. это приведет к сообщению об ошибке:

```
» 10 e9
???  10 e9
Missing operator, comma, or semi-colon.
```

Если требуется получить результат вычислений более точно, то следует выбрать в раскрывающемся списке long. Результат будет отображаться в длинном формате с плавающей точкой long с четырнадцатью цифрами после десятичной точки. Форматы short e и long e предназначены для вывода результата в экспоненциальной форме с четырьмя и пятнадцатью цифрами после десятичной точки соответственно. Информацию о форматах можно получить, набрав в командной строке команду help с аргументом format:

```
» help format
```

В командном окне появляется описание каждого из форматов. Задавать формат вывода можно непосредственно из командной строки при помощи команды format. Например, для установки длинного с плавающей точкой формата вывода результатов

вычислений следует ввести команду format long e в командной строке:

```
» format long e
» 1.25/3.11
ans =
4.019292604501608e-001
```

Обратите внимание, что команда help format выводит на экран название форматов прописными буквами. Однако команда, которую надо ввести, состоит из строчных букв. К этой особенности встроенной справки help надо привыкнуть. MatLab различает прописные и строчные буквы. Попытка набора команды прописными буквами приведет к ошибке:

```
» FORMAT LONG E
??? FORMAT LONG.
|
Missing operator, comma, or semi-colon.
```

Для более удобного восприятия результата MatLab выводит результат вычислений через строку после вычисляемого выражения. Однако иногда бывает удобно разместить больше строк на экране, для чего следует выбрать переключатель compact (File, Numeric display) из раскрывающегося списка. Добавление пустых строк обеспечивается выбором loose из раскрывающегося списка Numeric display.

Все промежуточные вычисления MatLab производит с двойной точностью, независимо от того, какой формат вывода установлен.

Использование элементарных функций

Предположим, что требуется вычислить значение следующего выражения:

$$e^{-2.5} \times (\ln 11.3)^{0.3} - \sqrt{(\sin 2.45\pi + \cos 3.78\pi)/(\mathrm{tg}\, 3.3)}\,.$$

Введите в командной строке это выражение в соответствии с правилами MatLab и нажмите <Enter>:

```
» exp(-2.5)*log(11.3)^0.3-sqrt((sin(2.45*pi)+cos(3.78*pi)}/tan(3.3))
```

Ответ выводится в командное окно:

```
ans =
-3.2105
```

При вводе выражения использованы встроенные функции MatLab для вычисления экспоненты, натурального логарифма, квадратного корня и тригонометрических функций. Какие встроенные элементарные функции можно использовать и как их вызывать? Наберите в командной строке команду help eifun, при этом в командное окно выводится список всех встроенных элементарных функций с их кратким описанием. Аргументы функций заключаются в круглые скобки, имена функций набираются строчными буквами. Для ввода числа л достаточно набрать pi в командной строке.

Арифметические операции в MatLab выполняются в обычном порядке, свойственном большинству языков программирования:

- возведение в степень ^;
- умножение и деление *, /;
- сложение и вычитание +, -.

Для изменения порядка выполнения арифметических операторов следует использовать круглые скобки. Если теперь требуется вычислить значение выражения, похожего на предыдущее, например

$$e^{-2.5} \times (\ln 11.3)^{0.3} + ((\sin 2.45\pi + \cos 3.78\pi)/(\mathrm{tg} 3.3))^2,$$

то необязательно его снова набирать в командной строке. Можно воспользоваться тем, что MatLab запоминает все вводимые команды. Для повторного занесения их в командную строку служат клавиши <up>, <down>.

Вычислите данное выражение, проделав следующие шаги.

1. Нажмите клавишу <>, при этом в командной строке появится введенное ранее выражение.
2. Внесите в него необходимые изменения, заменив знак минус на плюс и квадратный корень на возведение в квадрат (для перемещения по строке с выражением служат клавиши <right>, <left>, <Home>, <End>).
3. Вычислите измененное выражение, нажав <Enter>.

Получается

```
»exp(-2.5)*log(11.3)^0.3+((sin(2.45*pi)+cos(3.78*pi))/tan(3.3))^2
ans =
121.2446
```

Если необходимо получить более точный результат, то следует выполнить команду format long e, затем нажимать клавишу <> до тех пор, пока в командной строке не появится требуемое выражения, и вычислить его, нажав <Enter>.

```
» format long e
» exp(-2.5)*log(11.3)^0.3+((sin.(2.45*pi)+cos(3.78*pi))/tan(3.3))^2
ans =
1.212446016556763e+002
```

Вывести результат последнего найденного выражения в другом формате можно без повторного вычисления. Следует изменить формат командой short, а затем посмотреть значение переменной ans, набрав ее в командной строке и нажав <Enter>:

```
» format short
» ans
ans =
121.2446
```

В рабочей среде MatLab 6.x для вызова ранее введенных команд имеется удобное средство - окно Command History с историей команд. История команд содержит время и дату каждого сеанса работы с MatLab 6.x. Для активизации окна Command History необходимо выбрать вкладку с одноименным названием. Текущая команда в окне изображена на синем фоне. Если щелкнуть на какой-либо команде в окне левой кнопкой мыши, то данная команда становится текущей. Для ее выполнения в MatLab надо применить двойной щелчок мыши или выбрать строку с командой при помощи клавиш <up>, <down> и нажать клавишу <Enter>. Лишнюю команду можно убрать из окна. Для этого ее надо сделать текущей и удалить при помощи клавиши <Delete>. Можно выделить несколько идущих подряд команд при помощи комбинации клавиш <Shift>+<up>, <Shift>+<down> и выполнить их при помощи <Enter> или удалить клавишей <Delete>.

Выделение последовательно идущих команд можно производить левой кнопкой мыши с одновременным удерживанием клавиши <Shift>. Если команды не идут одна за другой, то для их выделения следует использовать левую кнопку мыши с удерживанием клавиши <Ctrl>.

При щелчке правой кнопкой мыши по области окна Command History появляется всплывающее меню. Выбор пункта Copy приводит к копированию команды в буфер Windows. При помощи Evaluate Selection можно выполнить отмеченную группу команд. Для удаления текущей команды предназначен пункт Delete Selection. Для удаления всех команд до текущей - Delete to Selection, для удаления всех команд - Delete Entire History.

При вычислениях возможны некоторые исключительные ситуации, например деление на ноль, которые в большинстве языков программирования приводят к ошибке. При делении положительного числа на ноль в MatLab получается inf (бесконечность), а при делении отрицательного числа на ноль получается -inf (минус бесконечность) и выдается предупреждение:

```
» 1/0
Warning: Divide by zero.
ans =
Inf
```

При делении нуля на нуль получается NaN (не число) и также выдается предупреждение:

```
» 0/0
Warning: Divide by zero.
ans =
NaN
```

При вычислении, например sqrt(-1), никакой ошибки или предупреждения не возникает. MatLab автоматически переходит в область комплексных чисел:

```
»sqrt(-1.0)
ans =
0 + l.0000i
```

Как узнать, какие встроенные элементарные функции можно использовать и как их вызывать? Наберите в командной строке команду help eifun, при этом в командное окно выводится список всех встроенных элементарных функций с их кратким описанием.

Общение пользователя с программой происходит с помощью команд, набираемых в командной строке. Командная строка системы MATLAB начинается со знака приглашения »(prompt), после которого можно записать любое алгебраическое выражение,

т.е. выражение, состоящее из чисел, имен переменных и знаков операций. Для того чтобы увидеть результат вычисления математического выражения, которое вводится после знака », нужно нажать клавишу Enter. После вычисления в нижней части экрана появится строка для ввода новых данных, которая также начинается со знака приглашения.

!! *Введите на поле нового документа 3+5 и нажмите Enter.*

Что сделала программа MATLAB? Сначала она вычислила сумму 3+5, затем записала результат в специальную системную переменнуюansи вывела ее значение, равное 8, в командное окно. Ниже ответа расположена командная строка с мигающим курсором, обозначающая, чтоMATLABготова к дальнейшим вычислениям. Можно набирать в командной строке новые выражения и находить их значения.

Если требуется продолжить работу с предыдущим выражением, например, вычислить (3+5)/4, то проще всего воспользоваться уже имеющимся результатом, который хранится в переменной ans.

!! *Наберите в командной строке ans/4 и нажмите Enter.*

!! *Присвойте переменной a полученный результат.*

Для того чтобы ввести в строке ввода значения нескольких переменных подряд, между выражениями ставится знак ";". Необходимо отметить, что знак ";" используется также для того, чтобы система не выводила на экран результат после нажатия клавиши Enter.

!! *Введите в одной строке b=5, c=10, k=2.*

Вся информация, которая набирается в строке ввода, запоминается в специальной области памяти, называемой стеком команд. С помощью нажатия клавиш управления курсором ↑или ↓ можно вернуть в строку ввода ранее набранные команды.

!! *Попробуйте просмотреть ранее введенные команды с помощью этих клавиш, а затем вернитесь к чистой командной строке.*

Расположение информации, которая отображена в рабочем окне MATLAB, можно условно разделить на две зоны: зону просмотра и зону ввода (или редактирования) информации. В зоне просмотра можно выделять (с помощью мыши) любую информацию и копиро-

вать ее в буфер обмена, однако в ней ничего нельзя исправлять. В свою очередь, зона ввода занимает последнюю строку рабочего окна. В ней можно вводить и редактировать набранные выражения.

Для просмотра рабочей области удобно использовать полосы скроллинга или клавиши Home, End для перемещения влево или вправо и PageUpиPageDown для перемещения вверх или вниз. Если вдруг после перемещения по рабочей области командного окна пропала командная строка с мигающим курсором, просто нажмите Enter.

Важно запомнить, что набор любой команды или выражения должен заканчиваться нажатием на Enter, для того, чтобы программа MATLAB выполнила эту команду или вычислила выражение.

Иногда работа в системе MATLAB существенно замедляется. Это происходит из-за того, что в специальной области памяти компьютера запоминается огромное количество значений всех переменных, вычисленных в течение текущего сеанса работы. Данная область называется рабочим пространством системы MATLAB. Для того чтобы скорость работы не замедлялась, переменные, которые уже не нужны для работы, необходимо стирать из памяти компьютера. Это делается с помощью команды clear, имеющей следующий синтаксис:

clear('имя1' ,'имя2' ,...) .

Указанная команда удаляет из рабочего пространства переменные с именами имя1, имя2 и т.д.

Выполнив команду who, можно узнать, какие переменные остались в рабочем пространстве.

!! *Выполните команду who.*

!! *Удалите из памяти переменные b и c.*

!! *Проверьте результат.*

Команда whos позволяет получить более подробную информацию о переменных в виде таблицы.

Первый столбик Name состоит из имен используемых переменных. То, что содержится в столбике Size, по существу, определяется основным принципом работы MATLAB. Программа MATLAB все данные представляет в виде массивов. Переменные

a1, a2 и a3 являются двумерными массивами размера один на один. Каждая из переменных занимает по восемь байтов, как указано в столбике Bytes. Наконец, в последнем столбике Class указан тип переменных — doublearray, т. е. массив, состоящий из чисел двойной точности. В строке под таблицей написано, что в итоге три элемента, т. е. переменные, занимают двадцать четыре байта. Оказывается, что представление всех данных в виде массивов дает определенные преимущества.

Удалить из памяти компьютера все ранее использованные переменные можно, выполнив команду clear без указания их имен и без скобок.

Команда clc очищает видимую часть рабочего пространства.

Для просмотра значения какой-либо переменной, содержащейся в рабочем пространстве, достаточно набрать ее имя и нажать кнопку Enter.

!! *Попробуйте просмотреть значение переменной b.*

Система MATLAB выведет сообщение о неопределенности переменной.

В MATLAB имеется другое удобное средство для просмотра переменных рабочей среды – окно Workspace, для перехода к которому следует активизировать одноименную закладку в меню Veiw.

!! *Откройте окно Workspace.*

Двойной щелчок по строке соответствующей, каждой переменной приводит к отображению её содержимого в отдельном окне, что особенно полезно при работе с массивами. Панель инструментов окна Workspace позволяет удалить лишние переменные, сохранить и открыть рабочую среду.

Форматы вывода результата вычислений

Встроенные математические функции MATLAB позволяют находить значения различных выражений. MATLAB предоставляет возможность управления форматом вывода результата. Команды для вычисления выражений имеют вид, свойственный всем языкам программирования высокого уровня.

Вид, в котором выводится результат вычислений, зависит от формата вывода, установленного в MATLAB. Далее объяснено, как задать основные форматы вывода.

Требуемый формат вывода результата определяется пользователем из меню MATLAB. Выберите в меню File пункт Preferences. На экране появится диалоговое окно Preferences, в котором следует перейти на вкладку CommandWindow.

Задание формата производится из раскрывающегося списка NumericFormat панели Textdisplay.

Разберем пока только наиболее часто используемые форматы. Выберите short в раскрывающемся списке NumericFormat. Закройте диалоговое окно, нажав кнопку ОК. Сейчас установлен короткий формат с плавающей точкой short для вывода результатов вычислений, при котором на экране отображаются только четыре цифры после десятичной точки.

!! *Наберите в командной строке 100/3 и нажмите Enter.*

Этот формат вывода сохранится для всех последующих вычислений, если только не будет установлен другой формат. Заметьте, что в MATLAB возможна ситуация, когда при отображении слишком большого или малого числа результат не укладывается в форматshot.

!! *Вычислите 10 000/3 и 1/3000*

Обратите внимание на то, что результат выводится в экспоненциальной форме.

Однако первоначальная установка формата сохраняется и при дальнейших вычислениях, для небольших чисел вывод результата снова будет происходить в формате shot.

В экспоненциальной форме можно самостоятельно набирать числа в выражениях. Например, проще набрать 10e9 или 1.0e10, чем 1 000 000 000, а результат будет тот же самый. Пробел между цифрами и символом e при вводе не допускается, т. к. это приведет к сообщению об ошибке.

Если требуется получить результат вычислений более точно, то в диалоговом окне Preferences следует выбрать в раскрывающемся списке NumericFormat панели Textdisplay формат long. Результат

будет отображаться в длинном формате с плавающей точкой long с четырнадцатью цифрами после десятичной точки.

!! *Проделайте те же вычисления после переустановки формата.*

Форматы shorte и longe предназначены для вывода результата в экспоненциальной форме с четырьмя и пятнадцатью цифрами после десятичной точки соответственно.

Задавать формат вывода можно непосредственно из командной строки при помощи команды format. Например, для установки длинного с плавающей точкой формата вывода результатов вычислений следует ввести команду formatlong в командной строке.

Команда, которую надо ввести, состоит из строчных букв. MATLAB различает прописные и строчные буквы. Попытка набора команды прописными буквами приведет к ошибке.

!! *Измените формат на short с помощью команды и проверьте её срабатывание с помощью вычисления одного из уже вычисленных выражений.*

Для более удобного восприятия результата MATLAB выводит результат вычислений через строку после вычисляемого выражения. Однако иногда бывает удобно разместить больше строк на экране, для чего следует в диалоговом окне Preferences на вкладке Command Window выбрать compact из раскрывающегося списка Numericdisplay. Добавление пустых строк обеспечивается выбором команды loose из раскрывающегося списка из того же раскрывающегося списка.

В рабочей среде MATLAB для вызова ранее введенных команд имеется дополнительное удобное средство — окно Command Historyс историей команд. История команд содержит время и дату каждого сеанса работы с MATLAB. Для активизации окна Command History необходимо выбрать вкладку с одноименным названием из менюView.

!! *Откройте окно Command History.*

Если щелкнуть на какой-либо команде в окне левой кнопкой мыши, то данная команда становится текущей и выделяется синим цветом. Для ее выполнения в MATLAB надо применить двойной

щелчок мыши. Лишнюю команду можно убрать из окна. Для этого ее надо сделать текущей и удалить при помощи клавиши Delete. Можно выделить несколько идущих подряд команд при помощи комбинации клавиш Shift+↑,Shift+↓и выполнить их при помощи Enter или удалить клавишей Delete. Выделение последовательно идущих команд можно производить левой кнопкой мыши с одновременным удерживанием клавиши Shift. Если команды не идут одна за другой, то для их выделения следует использовать левую кнопку мыши с удерживанием клавиши Ctrl.

При щелчке правой кнопкой мыши по области окна Command History появляется всплывающее меню. Выбор пункта Copy приводит к копированию команды в буфер Windows. При помощи Evaluate Selection можно выполнить отмеченную группу команд. Команда CreateM-file служит для создания нового М-файла. Для удаления текущей команды предназначен пункт DeleteSelection. Для удаления всех команд до текущей —DeletetoSelection, для удаления всех команд —DeleteEntireHistory.

!! *Удалите две последние команды.*

При удалении команды из окна Command History ее запись в Command Window не исчезает, однако, просмотреть ее в этом окне уже невозможно.

!! *Закройте окно Command History.*

!! *Закройте окно Workspace.*

Использование элементарных функций

При вводе вычисляемых выражений используются встроенные функции MATLAB для вычисления экспоненты, натурального логарифма, квадратного корня и тригонометрических функций и др. Аргументы функций заключаются в круглые скобки, имена функций набираются строчными буквами. Для ввода числаπдостаточно набрать рів командной строке.

Арифметические операции в MATLAB выполняются в обычном порядке, свойственном большинству языков программирования:

- возведение в степень ^;
- умножение и деление *, /;
- сложение и вычитание +, -.

Для изменения порядка выполнения арифметических операторов следует использовать круглые скобки.

При вычислениях возможны некоторые исключительные ситуации, например деление на ноль, которые в большинстве языков программирования приводят к ошибке. При делении положительного числа на ноль в MATLAB получается Inf(бесконечность), а при делении отрицательного числа на ноль получается -Inf (минус бесконечность).

При делении нуля на ноль получается Nan(не число).

!! *Попробуйте проделать описанные действия с нулем. Обратите внимание на результат.*

При вычислении, например $\sqrt{-1}$, никакой ошибки или предупреждения не возникает. MATLAB автоматически переходит в область комплексных чисел.

Вычисления с комплексными числами

При наборе комплексных чисел в командной строке MATLAB можно использовать либо i, либо j, сами числа при умножении, делении и возведении в степень необходимо заключать в круглые скобки. Если не использовать скобки, то умножаться или возводиться в степень будет только мнимая часть и получится неверный результат.

Для вычисления комплексно-сопряженного числа применяется апостроф, который следует набирать сразу за числом, без пробела.

Если необходимо найти комплексно-сопряженное выражение, то исходное выражение должно быть заключено в круглые скобки

MATLAB позволяет использовать комплексные числа в качестве аргументов встроенных элементарных функций.

Функции для работы с комплексными числами

К ним относятся следующие функции MATLAB:

- abs и angle — модуль r и фаза φ (в радианах от -π до π) комплексного числа, a + i•b = r• (cos φ+ i • sinφ);
- complex(a,b) — конструирует комплексное число по его действительной и мнимой части:
- conj— возвращает комплексно-сопряженное число;

- imag,real— возвращает мнимую и действительную часть комплексного числа.

!! *Возведите число* 1-i·2 *в третью степень. Присвойте это значение переменной x.*

!! *Найдите все значения корня третьей степени из этого числа. Присвойте полученные значения x1, x2, x3.*

!! *Найдите комплексно-сопряженное ему выражение.*

Встроенные элементарные функции

Встроенные элементарные функции MATLAB включают тригонометрические, гиперболические, экспоненциальные и логарифмические функции, а также функции для работы с комплексными числами и для округления различными способами.

Для того, чтобы узнать, какие встроенные элементарные функции имеются в MATLAB необходимо набрать в командной строке команду helpelfun. При этом в командное окно выводится список всех встроенных элементарных функций с их кратким описанием.

!! *Просмотрите все имеющиеся элементарные функции.*

!! *Введите в командной строке это выражение в соответствии с правилами MATLAB и вычислите его значение.*

$$e^{-2.5}\cdot(\ln 11.3)^{0.3}+\left(\frac{\sin 2.45\cdot\pi+\cos 3.78\cdot\pi}{tg3.3}\right)^2$$

Аргументы тригонометрических функций должны быть выражены в радианах. Обратные тригонометрические функции возвращают результат также в радианах.

Использование переменных

Как и во всех языках программирования, в MATLAB предусмотрена возможность работы с переменными. Причем пользователь не должен заботиться о том, какие значения будет принимать переменная (комплексные, вещественные или только целые). Для того чтобы присвоить, например, переменной z значение 1.45, достаточно написать в командной строкеz=1.45, при этом MATLAB сразу же выведет значение z.

Здесь знак равенства используется в качестве оператора присваивания. Часто не очень удобно после каждого присваивания

получать еще и результат. Поэтому в MATLAB предусмотрена возможность завершать оператор присваивания точкой с запятой для подавления вывода результата в командное окно. Именем переменной может быть любая последовательность букв и цифр без пробела, начинающаяся с буквы. Строчные и прописные буквы различаются. Количество воспринимаемых MATLAB символов в имени переменной составляет 31 (такие длинные имена и не нужны).

!! *Найдите значение следующего выражения, применяя введение промежуточных переменных*

$$\frac{\dfrac{\sin 1.3\pi}{\ln 3.4}+\sqrt{\dfrac{tg\,2.75}{th2.75}}}{\dfrac{\sin 1.3\pi}{\ln 3.4}-\sqrt{\dfrac{tg\,2.75}{th2.75}}}$$

MATLAB запоминает значения всех переменных, определенных во время сеанса работы. Если после ввода примера, приведенного выше, были проделаны еще какие-либо вычисления, и возникла необходимость вывести значение одной из введенных переменных, то следует просто набрать имя переменной в командной строке и нажать Enter.

Переменные, определенные выше, можно использовать и в других формулах.

Для ввода длинных формул или команд в командную строку следует поставить три точки (подряд, без пробелов), нажать клавишу Enter и продолжить набор формулы на следующей строке. Так можно разместить выражение на нескольких строках. MATLAB вычислит все выражение или выполнит команду после нажатия на Enter в последней строке (в которой нет трех идущих подряд точек).

! *Найдите значение этого же выражения, записав его в одну строку.*

Предположим, что часть вычислений с переменными выполнена, а остальные придется доделать во время следующего сеанса работы с MATLAB. В этом случае понадобится сохранить переменные, определенные в рабочей среде.

Сохранение рабочей среды

Самый простой способ сохранить значения всех переменных — использовать в меню File пункт SaveWorkspaceAs. При этом появляется диалоговое окно SaveWorkspaceVariable, в котором следует указать каталог и имя файла. По умолчанию предлагается сохранить файл в подкаталоге work основного каталога MATLAB.

Предлагаемое расширение при сохранении единственное mat.

!! *Сохраните свой файл под именем Пример.*

Работа с массивами

Все данные MATLAB представляет в виде массивов.

Массив — упорядоченная, пронумерованная совокупность однородных данных. Создавая массив, вместо того, чтобы давать каждой ячейке памяти, используемой для хранения одного элемента данных, отдельное имя, всей совокупности данных дается одно имя.

Массивы различаются по числу размерностей или измерений: одномерные, двумерные, многомерные. Размером массива называют число элементов, вдоль каждого из измерений. Доступ к элементам осуществляется при помощи индекса. В MATLAB нумерация элементов массивов начинается с единицы. Это значит, что индексы должны быть больше или равны единице.

Вектор, вектор-строка, матрица или тензор являются математическими объектами, а одномерные, двумерные или многомерные массивы — способы хранения этих объектов в компьютере. Вектор может быть записан в столбик (вектор-столбец) и в строку (вектор-строка). Вектор-столбцы и вектор-строки будут называться просто векторами.

Одномерные массивы

Для формирования массива используют операцию конкатенации, которая обозначается квадратными скобками.

!! *Введите на лист a=[10,4,6] и нажмите Enter.*

В результате её выполнения сформировался массив чисел.

Числовые массивы являются элементами типа double.

В качестве элементов массива могут использоваться любые переменные типа double, т.е. вещественные или комплексные числа, а также переменные, которые сами являются массивами. Для доступа к конкретному элементу массива используется операция индексации, которая обозначается круглыми скобками.

!! *Введите на лист a(1) и нажмите Enter.*

Для того, чтобы присвоить элементу массива новое значение, к нему надо применить одновременно операции индексации и присваивания.

!! *Присвойте второму элементу массива a значение -5.*

Массив можно расширить, если присвоить несуществующему элементу массива какое-нибудь значение. Все отсутствующие элементы массива до того, которому значение присвоено, примут значение ноль.

!! *Присвойте шестому элементу массива a значение 6.*

Выполнив теперь функцию length(имя), можно узнать, из скольких элементов состоит массив с указанным именем.

!! *Узнайте число элементов массива a.*

Рассмотрим другой способ создания массивов. А именно с помощью функций ones и zeros, которые сразу создают массив нужного размера, заполненный, соответственно, единицами (ones) и нулями (zeros). Аргументами для этих функций служат размеры массива, начиная с количества строк, указанные через запятую. Затем с помощью операций индексации и присваивания созданный массив заполняется нужными элементами.

!! *Создайте массив b, имеющий такую же размерность, что и массив a с помощью функции ones или zeros. Заполните его действительными элементами.*

При необходимости создать массив, состоящий из чисел, изменяющихся с заданным шагом по мере увеличения индекса, используется операция “:”. Полученная команда будет выглядеть так:

$c=c_{нач}:h:c_{кон}$,

где $c_{нач}$– начальное значение элементов массива, h– шаг изменения, $c_{кон}$– конечное значение элементов массива.

!! *Создайте массив c, имеющий такую же размерность, что и массивы a и b, элементы которого изменяются с шагом 0.5, начиная с 3.*

Необязательно заботиться о том, чтобы сумма предпоследнего значения и шага равнялась бы конечному значению, т.к. массив будет заполняться до тех пор, пока не будет достигнуто такое значение, которое будет равно конечному или отличается от него меньше, чем на шаг.

В случае отрицательного шага для получения непустой вектор-строки начальное значение должно быть больше конечного.

Шаг, равный единице, допускается не указывать при автоматическом заполнении.

Применение функций обработки данных к векторам

Перемножение элементов вектора или вектора-строки осуществляется при помощи функции prod(имя).

Функция sum предназначена для суммирования элементов вектора.

!! *Вычислите среднее арифметическое элементов вектора c.*

!! *Проверьте результат, вычислив среднее арифметическое используя встроенную функцию mean.*

Для нахождения минимума и максимума из элементов вектора служат встроенные функции min и max.

Часто необходимо знать не только значение минимального или максимального элемента в массиве, но и его индекс (порядковый номер). Для того, чтобы зафиксировать порядковый номер найденного элемента необходимо записать вызываемую функцию в следующем виде [m,k]=min(имя) или[m,k]=max(имя). Тогда m будет максимальным или минимальным элементом в массиве, a_k– его порядковым номером.

!! *Найдите максимальный элемент в массиве a, а также его порядковый номер.*

В число основных функций для работы с векторами входит функция упорядочения вектора по возрастанию его элементов sort.

!! *Отсортируйте по возрастанию вектор a.*

!! *Попробуйте упорядочить вектор по убыванию, используя эту же функцию sort.*

Упорядочение элементов в порядке возрастания их модулей производится с привлечением функции abs.

Вызов sort с двумя выходными аргументами приводит к образованию массива индексов соответствия элементов упорядоченного и исходного массивов.

Если аргументом функций max и min является вектор, состоящий из комплексных чисел, то результатом является максимальный или минимальный по модулю элемент. Функция sort также упорядочивает комплексный вектор по модулю, а компоненты с равными модулями располагаются в порядке возрастания фаз.

!! *Отсортируйте вектор*

d=$\overline{(\ 2+3i\ \ -0.5-i\ \ i\ \ -2.5+0.6i\ \ -2+9i\ )}$

по возрастанию вместе с индексами.

В число встроенных функций входит дискретное преобразование Фурье -fft, свертка —conv, работа со звуком —sound и многие другие.

Поэлементные операции с векторами

Если вектор используется в качестве аргумента математической функции, результатом которой является вектор с элементами, равными значениям функции от соответствующих элементов исходного вектора, то происходит, так называемое, поэлементное вычисление вызываемой функции. В этом разделе описаны возможности поэлементной работы с векторами, которые понадобятся в дальнейшем для определения некоторых функций и построения их графиков.

!! *Введите две вектор-строки: v1=[2 -3 4 1]; v2=[7 5 -6 9].*

Операция .* (не вставляйте пробел между точкой и звездочкой!) приводит поэлементному умножению векторов одинаковой длины. В результате получается вектор с элементами, равными произведению соответствующих элементов исходных векторов.

!! *Осуществите поэлементное умножение v1 и v2.*

При помощи «^» осуществляется поэлементное возведение в степень.

Показателем степени может быть вектор той же длины, что и возводимый в степень. При этом каждый элемент первого вектора возводится в степень, равную соответствующему элементу второго вектора:

!! *Осуществите поэлементное возведение в степень v2 вектора v1.*

Деление соответствующих элементов векторов одинаковой длины выполняется с использованием операции ./

Обратное поэлементное деление (деление элементов второго вектора на соответствующие элементы первого) осуществляется при помощи операции «\».

!! *Осуществите поэлементное прямое и обратное деление v1 и v2.*

Точка в MATLAB используется не только для ввода десятичных дробей, но и для указания того, что деление или умножение массивов одинакового размера должно быть выполнено поэлементно.

К поэлементным относятся и операции с вектором и числом. Сложение вектора и числа не приводит к сообщению об ошибке. MATLAB прибавляет число к каждому элементу вектора. То же самое справедливо и для вычитания.

!! *Прибавьте к каждому элементу v2 число 5.*

Умножать вектор на число можно как справа, так и слева.

!! *Умножьте каждый элемент вектора v1 на -1.*

Делить при помощи знака / можно вектор на число. Попытка деления числа на вектор приводит к сообщению об ошибке.

Это связано с тем, что операция / в MATLAB предназначена, в частности, для: решения систем линейных алгебраических уравнений. Если требуется разделить число на каждый элемент вектора и записать результат в новый вектор, то следует использовать операцию ./

Все вышеописанные операции применимы как к вектор-строкам, так и к вектор-столбцам.

Разберем, как правильно транспонировать и вычислять сопряженные векторы в MATLAB для векторов, состоящих из комплексных элементов.

Для нахождения сопряженного вектора в MATLAB используется апостроф, а для транспонирования следует применять точку с апострофом.

!! *Найдите сопряженный и транспонированный вектора для d.*

Операции .' и ' над вещественными векторами приведут к одинаковым результатам. Поэлементные вычисления с массивами используются на протяжении всей книги.

Ясно, что для векторов, состоящих только из действительных чисел, операции сопряжения и транспонирования совпадают.

Сложение и вычитание векторов

При сложении векторов используется тот же знак плюс, что и для сложения чисел. Естественно, для нахождения разности векторов следует применять знак минус. Результатом операции будет вектор того же размера, что и исходные, каждый элемент которого будет равен сумме или разности соответствующих элементов исходных векторов.

Если размеры векторов, к которым применяется сложение или вычитание, не совпадают, то выдается сообщение об ошибке.

Умножение векторов

Вектор можно умножить на другой вектор скалярно (это произведение еще называют внутренним), векторно, или образовать так называемое внешнее произведение. Результатом скалярного произведения является число, векторного — вектор, а внешнего — матрица.

Скалярное произведение

Скалярное произведение векторов a и b длины N, состоящих из действительных чисел, определяется формулой

$$a \cdot b = \sum_{k=1}^{N} a_k \cdot b_k$$

Следовательно, для вычисления скалярного произведения необходимо просуммировать компоненты вектора, полученного в результате поэлементного умножения a на b, т. е. надо использовать функцию sum и поэлементное умножение.

!! *Найдите самостоятельно скалярное произведение векторов: a=[1.2 -3.2 0.7] и b=[4.1 6.5 -2.9]*

!! *Найдите длину (модуль) вектора a.*

Векторное произведение

Векторное произведение a*b определено только для векторов из трехмерного пространства, т. е. состоящих из трех элементов. Результатом также является вектор из трехмерного пространства. Для вычисления векторного произведения в MATLAB служит функция cross.

!! *Вычислите векторное произведение векторов a и b.*

!! *Попробуйте вычислить axb + bxa.*

Смешанное произведение

Смешанное произведение векторов a, b, c определяется по формуле abc = a·(b*c). Модуль смешанного произведения векторов равен объему параллелепипеда, построенного на этих векторах

!! *Найдите объем параллелепипеда, построенного на векторах a, b и c=[2.8 -0.2 -1.9]*

Внешнее произведение

Внешним произведением векторов $a=(a_j)_{j=1,\ldots,N}$,$b=(b_k)_{k=1,\ldots,M}$ называется матрица $c=(c_{jk})_{j=1,\ldots N,\ k=1,\ldots,M}$ размера NxM, элементы которой вычисляются по формуле

$c_{jk}=a_j \cdot b_k$.

Вектор-столбец a в MATLAB представляется в виде двумерного массива размера N на один. Вектор-столбец b при транспонировании переходит в вектор-строку размера один на M. Вектор-столбец и вектор-строка есть матрицы, у которых один из размеров равен единице. Фактически, $c=ab^T$, где умножение происходит по правилу матричного произведения. Для вычисления матричного произведения в MATLAB используется оператор "звездочка".

!! *Найдите внешнее произведение векторов a и b.*

Двухмерные массивы

Массивы такого типа подобны одномерным, за исключением того, что их элементы определяются не одним индексом, а двумя. В математике подобные массивы называют матрицами, состоящими из строк и столбцов. Любая строка (или столбец) в матрице является одномерным массивом, который принято называть вектор-строкой или вектор-столбцом соответственно. Формирование матрицы осуществляется операцией конкатенации, которая обозначается внешними квадратными скобками. При вертикальной конкатенации элементы каждой последующей строки массива отделяются от предыдущей точкой с запятой, в то время как элементы одной и той же строки разделяются запятыми либо пробелами.

!! *Задайте матрицу*

$$l = \begin{pmatrix} 4 & 5 \\ 7 & 9 \\ 3 & 1 \end{pmatrix}$$

используя вертикальную конкатенацию.

Эту же матрицу можно сформировать горизонтальной конкатенацией вектор-столбцов. При горизонтальной конкатенации элементы каждого столбца заключаются в собственные квадратные скобки и между собой разделяются знаком “;”, а элементы разных столбцов разделяются запятой.

!! *Задайте матрицу*

$$m = \begin{pmatrix} -2 & 5 \\ 1 & -6 \\ 0 & 1 \end{pmatrix}$$

используя горизонтальную конкатенацию.

Элементы матрицы можно также задавать с помощью функции cat, аргументы которой заключаются в круглые скобки. Для вертикальной конкатенации ее первый аргумент равен 1, а для горизонтальной - равен 2. *l*=cat(1,[4,5],[7,9],[3,1]),m=cat(2,[-2,1,0],[5,-6,1]).

!! *Задайте матрицу*

$$k=\begin{pmatrix} 6 & -6 \\ -2 & 2 \\ 3 & 0 \end{pmatrix}$$

используя вертикальную конкатенацию.

Есть еще один способ задания матрицы – ввод матрицы построчно. Имени матрицы присваиваются после открывающей квадратной скобки элементы одной строки, затем нажимается Enter. Вводятся элементы второй строки, нажимается Enter. После введения последней строки ставится закрывающая квадратная скобка. Ввод матрицы закончен.

Размерность массива определяется с помощью функции ndims. Размер созданного массива можно узнать с помощью функции size.

Результатом функции ndims является число, указывающее количество направлений, вдоль которых расположены элементы массива. Результатом функции size являются числа, причем первое из них - количество строк, второе — количество столбцов.

!! *Определите размерность и размер одного из созданных массивов.*

!! *Любым способом определите имена переменных, находящихся на данный момент в Рабочем пространстве.*

!! *Примените команду size к переменной, которая состоит из одного числа.*

Отсюда видно, что в системе MATLAB все переменные типа double представляются в виде двухмерных массивов, а именно: векторы - в виде двухмерных массивов, размер которых по одному из направлений равен единице; матрицы - в виде двухмерных массивов размера nxn; скаляры - в виде двухмерных массивов размером 1x1.

Существует также пустой массив, обозначаемый квадратными скобками [ ], между которыми ничего нет. Такой массив трактуется как матрица размером 0x0. Обычно пустой массив используют для того, чтобы удалять строки или столбцы матриц. Например, для того, чтобы удалить в матрице m третью строчку можно набрать команду m(3,:)=[]. Символ“:”означает все элементы строки (столбца).

!! *Выполните эту команду.*

Доступ к элементам матриц осуществляется при помощи двух индексов — номеров строки и столбца, заключенных в круглые скобки. Элементы матриц могут входить в состав выражений.

Расположение элементов матрицы в памяти компьютера определяет еще один способ обращения к ним. Матрица a размера n на m хранится в виде вектора длины n·m, в котором элементы матрицы расположены один за другим построчно. Для доступа к элементам матрицы можно использовать один индекс, задающий порядковый номер элемента матрицы в векторе.

!! *Просмотрите элемент матрицы k(5).*

Блочные матрицы

Очень часто в приложениях возникают так называемые блочные матрицы, т. е. матрицы, составленные из непересекающихся подматриц (блоков). Соответствующие размеры блоков должны совпадать.

!! *Введите матрицы*

$$A := \begin{pmatrix} -1 & 4 \\ -1 & 4 \end{pmatrix} \quad B := \begin{pmatrix} 2 & 0 \\ 0 & 5 \end{pmatrix} \quad C := \begin{pmatrix} 3 & -3 \\ -3 & 3 \end{pmatrix} \quad D := \begin{pmatrix} 8 & 9 \\ 1 & 10 \end{pmatrix}$$

Требуется составить блочную матрицу K, считая, что она состоит из двух строк, в первой – A и B, а во второй –C и D.

Можно было поступить и по-другому, а именно, считать, что матрица K состоит из двух столбцов, в первом — матрицы A и C, а во втором — B и D.

!! *Составьте блочную матрицу K.*

$$K := \begin{pmatrix} A & B \\ C & D \end{pmatrix}$$

Обратной задачей к конструированию блочных матриц является выделение блоков.

Выделение блоков матриц осуществляется индексацией при помощи двоеточия, которое ставится между номерами строк и столбцов, составляющих блок

!! *Выделите из матрицы K блок*

$$\begin{pmatrix} 4 & 0 \\ -3 & 8 \end{pmatrix}$$

Для выделения из матрицы столбца или строки (то есть массива, у которого один из размеров равен единице) следует в качестве одного из индексов использовать номер столбца или строки матрицы, а другой индекс заменить на двоеточие без указания пределов.

!! *Выделите из матрицы K последнюю строку.*

При выделении блока до конца матрицы можно не указывать ее размеры, использовать элемент end.

!! *Выделите из матрицы K вторую строку, начиная со второго элемента.*

!! *Создайте квадратную матрицу размера четыре, состоящую из случайных целых чисел от нуля до десяти, вычтите из нее матрицу, состоящую из пятерок.*

Функции для создания векторов и матриц

1. eye(n) - формирование единичной матрицы размера n$\times$n.

2. diag(v,n) - формирование диагональной матрицы диагональной матрицы из элементов заданного вектора v с дополнительным числом строк и столбцов n. Число n может быть как положительным (строки и столбцы добавляются спереди и снизу), так и отрицательным (строки и столбцы добавляются сверху и справа).

3. rand(n) - формирование матрицы размера n$\times$n случайных элементов.

4. ones(n) - формирование матрицы единичных элементов размера nxn.

5. zeros(n) - формирование матрицы размера n$\times$n нулевых элементов.

6. magic(n) - формирование магической матрицы размера n$\times$n.

7. pascal(n) - формирование матрицы Паскаля размера n$\times$n.

8. toeplitz(n) - формирование матрицы Теплица размера n$\times$n.

9. Wilkinson(n) - формирование матрицы Уилкинсона размера n$\times$n.

Функции над квадратной матрицей.

1. diag(A) – определение элементов главной диагонали матрицы A.

2. fliplr(A) – формирование матрицы с элементами, переставленными относительно вертикальной оси матрицы A.
3. flipud(A) - формирование матрицы с элементами, переставленными относительно горизонтальной оси матрицы A.
4. prod(A) - формирование вектора произведений элементов каждого столбца матрицы A.
5. sum(A) - формирование вектора сумм элементов каждого столбца матрицы A.
6. rot90(A) - поворот матрицы A на 90° по направлению против часовой стрелки.
7. tril(A) - выделение нижней треугольной части матрицы A.
8. triu(A) - выделение верхней треугольной части матрицы A.

Функции линейной алгебры

1. cond(A) - возвращает число обусловленности, основанное на второй норме.
2. condeig(A) - формирует вектор чисел обусловленности для собственных значений матрицы A.
3. det(A) - вычисляет определитель матрицы A.
4. rank(A) - определяет ранг матрицы A.
5. norm(A) - вычисляет норму матрицы A.
6. orth(A) - формирует ортонормальный базис матрицы A.
7. null(A) - возвращает ортонормальный базис нулевого пространства.
8. trace(A) - вычисляет "след" матрицы A, т.е. сумму ее диагональных элементов.
9. eig(A) - возвращает вектор собственных значений матрицы A.
10. max(A) - определяет максимальные элементы в столбцах матрицы A.
11. min(A) - находит минимальные элементы в столбцах матрицы A.
12. hess(A) - формирует верхнюю форму Хессенберга.
13. schur(A) - формирует матрицу Шура.
14. poly(A) - возвращает вектор коэффициентов характеристического полинома матрицы A.
15. sort(A) - сортирует в порядке возрастания элементы каждого столбца матрицы A.
16. find(A>c) - возвращает набор индексов, удовлетворяющих заданному условию, т.е. определяет, какие элементы матрицы A больше указанного числа c, и возвращает соответствующие этим элементам индексы.

Часть 2. Изучение интерфейса для анализа систем управления по передаточным функциям.

Рассмотрим систему автоматического управления (САУ), описываемую линейным (линеаризованным) дифференциальным уравнением вида:

$$a_n \frac{d^n y(t)}{dt^n} + a_{n-1} \frac{d^{n-1} y(t)}{dt^{n-1}} + \ldots + a_1 \frac{dy(t)}{dt} + a_0 y(t) = \\ = b_m \frac{d^m u(t)}{dt^m} + b_m \frac{d^{m-1} u(t)}{dt^{m-1}} + \ldots + b_1 \frac{du(t)}{dt} + b_0 u(t), \quad , \quad (1)$$

где $u(t)$ – входной процесс, $y(t)$ – выходной процесс, a_i, b_j, ($i = \overline{0,n}$, $j = \overline{0,m}$) – постоянные коэффициенты, n, m ($n \le m$) – постоянные числа. В операторной форме выражение (1) может быть записано – $A(D)y(t) = B(D)u(t)$. Здесь D – оператор дифференцирования $\left(D \overset{def}{=} \frac{d}{dt}\right)$. Отсюда преобразование «вход-выход» системы представляется как:

$$\frac{y(t)}{u(t)} = \frac{B(D)}{A(D)} = W(D), \quad (2)$$

где $W(D)$ называется операторной передаточной функции.

Один из способов моделирования систем заключается в представлении преобразования «вход-выход» в виде комплексной передаточной функции:

$$\frac{y(s)}{u(s)} = \frac{B(s)}{A(s)} = W(s), \quad (3)$$

которая получается путем применения преобразования Лапласа к (2) при известных начальных нулевых условиях. Здесь s-комплексная переменная. Связь между операторной (2) и комплексной (3) передаточными функциями можно записать в виде: $W(s) = W(D)\big|_{D=s}$..

Комплексные числа, являющиеся корнями многочлена $B(s)$, называются нулями передаточной функции, а корни многочлена $A(s)$ – полюсами.

Явный вид связи входа и выхода определяется выражением:

$$y(t) = \int_0^t w(t)v(t-\tau)d\tau \quad , \qquad (4)$$

где $w(t)$ – оригинал (т.е. полученный с помощью обратного преобразования Лапласа) комплексной передаточной функции $W(s)$.

Динамические свойства систем характеризуют реакции на входные воздействия специального вида. В частности анализ выхода системы на единичный скачок и □ -функцию (дельта -функцию).

Пусть $u(t) = 1(t)$, то есть на вход системы подается функция Хевисайда (единичный скачок), определяемая как:

$$1(t) = \begin{cases} 0, & \text{при } t \le 0, \\ 1, & \text{при } t > 0. \end{cases}$$

График функции Хевисайда приведен на рисунке 2. Реакция системы управления на единичный скачек называется переходной функцией системы и обозначается $h(t)$.

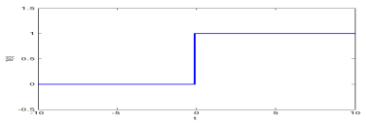


Рис.2 Функция Хевисайда.

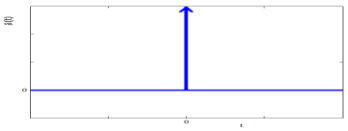


Рис. 3. Функция Дирака.

Если на вход системы поступает функция Дирака определяемая как:

$$\delta(t) = \begin{cases} \infty, & \text{при } t = 0, \\ 0, & \text{при } t \ne 0, \end{cases} \quad ,$$

то реакция САУ называется импульсной переходной функцией системы и обозначается $w(t)$. Таким образом оригинал комплексной передаточной функции можно измерить как реакцию систему на импульс.

Импульсная и переходная функции системы связаны соотношением (из (4)):

$$h(t) = \int_0^t w(\tau)d\tau$$

.

Пусть на вход системы с передаточной функцией $W(s)$ подается гармонический сигнал $u(t) = a_u \cos(\omega t)$, $t > 0$. (5). В этих условиях справедлива следующая теорема: если звено является устойчивым, то установившаяся реакция $y(t)$ на гармоническое воздействие является функцией той же частоты с амплитудой $a_y = a_u |W(i\omega)|$, и относительным сдвигом по фазе $\varphi = \arg(W(i\omega))$

Частотной характеристикой $W(i\omega)$ стационарной динамической системы называется преобразование Фурье переходной функции:

$$W(i\omega) = F[h(t,\tau)] = \int_0^\infty w(t-\tau)e^{-i\omega(t-\tau)}d\tau$$

,

где $w(t-\tau)$ – импульсная переходная функция.

Связь между комплексной передаточной функцией и частотной характеристикой, исходя из свойств преобразований Фурье можно представима в виде соотношения: $W(s)\big|_{s=i\omega} = W(i\omega)$. При фиксированном значении частотная характеристика является комплексным числом, и, следовательно, может быть представлена в виде:

$$W(i\omega) = A(\omega)e^{i\omega+\psi(\omega)} = U(\omega) + iV(\omega)$$

.

Здесь:

$A(\omega) = |W(i\omega)|$ – амплитудно-частотная характеристика (АЧХ);

$\psi(\omega) = \arg W(i\omega)$ – фазово-частотная характеристика (ФЧХ);

$U(\omega) = \mathrm{Re}\, W(i\omega)$ – вещественная частотная характеристика (ВЧХ);

$V(\omega) = \mathrm{Im}\, W(i\omega)$ – мнимая частотная характеристика (МЧХ).

Геометрическое место точек $W(iw)$ на комплексной плоскости при изменении частоты w от w_0 до от w_1 (обычно $w_0 = 0$, $w_1 = \infty$), называется амплитудно-фазовой характеристикой (АФХ) или частотным годографом Найквиста. Имеет широкое практическое

значение диаграмма Боде (логарифмическая амплитудная характеристика, ЛАХ), которая определяется как L=20lg(A(w)) измеряется в децибелах и строится как функция от lg(w).

Последовательность выполнения работы на ЭВМ (для части 1)

При использовании средств вычислительной техники работа выполняется в среде Matlab. Выполнение работы состоит из следующих шагов:

1. Запуск системы MATLAB.
2. Создание tf-объекта, в соответствии с вариантом задания.
3. Составляется дифференциальное уравнение, определяющее функционирование САУ.
4. Определяются полюса передаточной функции $s_i^*, (i=\overline{1,n})$ с использованием команды roots или pole.
5. Определяются нули передаточной функции $s_j^0, (j=\overline{1,m})$ с использованием команды roots или zero.
6. Используя LTI-viewer, или соответствующие команды строятся динамические характеристики – переходную функцию $h(t)$, импульсно-переходную функцию $w(t)$ и частотные характеристики – диаграмму Боде, частотный годограф Найквиста.
7. Исходной передаточная функция представляется в виде произведения типовых звеньев.

Примечание: Варианты заданий, состоят из двух цифр: первая - номер передаточной функции, вторая – номер набора значений коэффициентов и представляют собой последние две цифры номера студенческого билета (или задаются иным образом преподавателем).

Методический пример.

Задана передаточная функция САУ $W=\dfrac{s+2}{3s^3+4s^2+5s+3}$.

1. Создается LTI-объект с именем w, - выполняется:

```
>> w=tf([1 2],[3 4 5 3])

Transfer function:
        s + 2
----------------------
3 s^3 + 4 s^2 + 5 s + 3
```

2. Определяются полюса и нули передаточной функции с использованием команд pole, zero.

```
>> pole(w)

ans =

   -0.2639 + 1.0825i
   -0.2639 - 1.0825i
   -0.8055

>> zero(w)

ans =

     -2
```

3. Строится переходную функцию командой step(w). Результат приведен на рисунке 4.

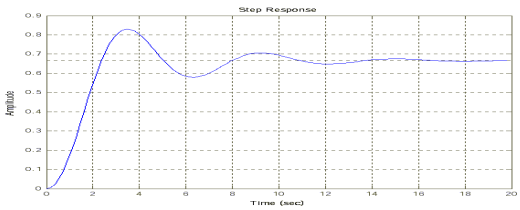


Рис. 4. Переходная функция h(t).

4. Строится импульсная переходная функция командой impulse(w). Результат показан на рисунке 5.

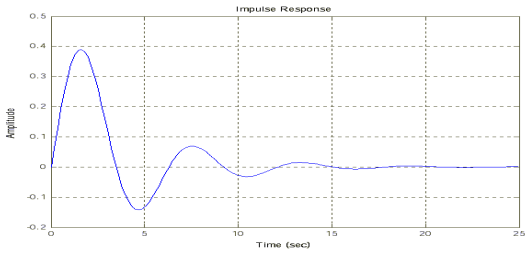


Рис. 5. -Импульсная переходная функция.

5. Строится диаграмма Боде, - команда bode(w) – рисунок 6.

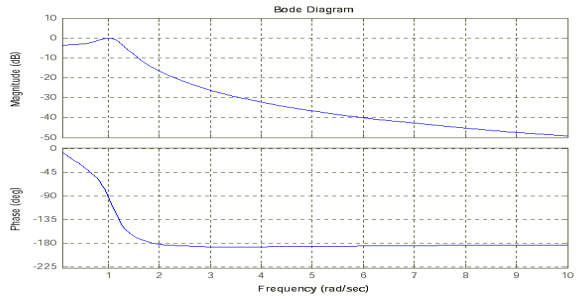


Рис. 6. Диаграмма Боде.

6. Строится частотный годограф Найквиста, - команда nyquist(w) – рисунок 7.

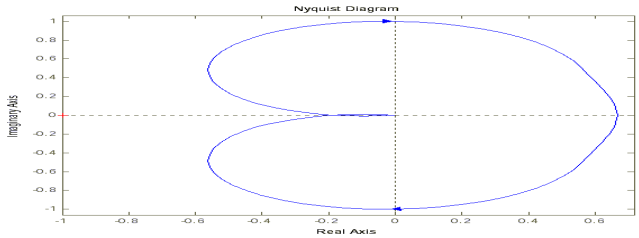


Рис. 7. Частотный годограф.

Аналогичные результаты (рисунок 8) можно получить, используя команду ltiview(w), с соответствующими настройками в меню «Plot Configuration».

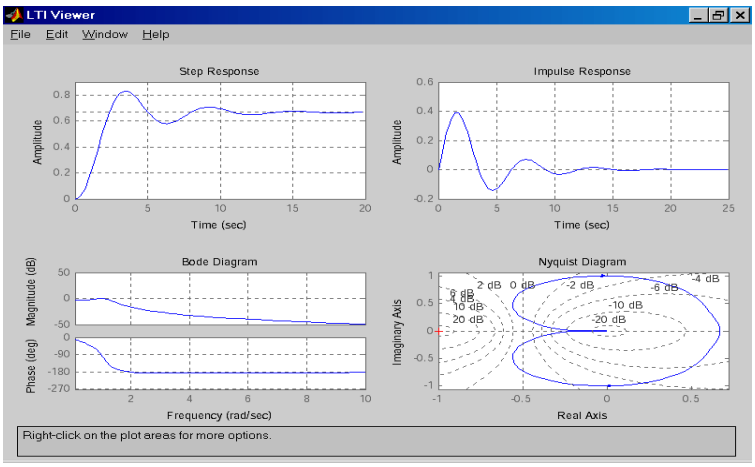


Рис. 8. – Результаты выполнения команды LTI-viewer.

Каждая из построенных характеристик полностью и однозначно определяет рассматриваемую систему управления.

Контрольные вопросы:

1. Представьте систему в виде последовательного соединения типовых звеньев.
2. Дайте определение и поясните физический смысл переходной функции.
3. Представьте исходную систему в пространстве состояний.
4. Найдите передаточную функцию замкнутой системы.
5. Постройте динамические характеристики типовых звеньев.
6. Определите вид ЛЧХ для пропорционально – интегрально – дифференциального регулятора (выбирается преподавателем).

Варианты заданий

№	Вид передаточной функции	№	Коэффициенты полиномов						
1.	$W(p)=\dfrac{b_1p+b_0}{a_4p^4+a_3p^3+a_2p^2+a_1p+a_0}$	1.	0	3	1	2	3	0	1
		2.	2	6	4	0	1	5	1
		3.	0	3	5	2	0	2	1
		4.	4	2	3	4	5	3	1
		5.	0	1	2	2	3	2	0
2.	$W(p)=\dfrac{b_2p^2+b_1p+b_0}{a_3p^3+a_2p^2+a_1p+a_0}$	1.	0	3	2	4	2	3	9
		2.	8	0	3	4	6	4	-1
		3.	4	6	2	5	5	0	1
		4.	6	8	7	0	6	3	-1
		5.	2	-1	-3	1	0	7	-2
3	$W(p)=\dfrac{b_2p^2+b_1p+b_0}{a_4p^4+a_3p^3+a_1p+a_0}$	1.	0	2	8	-3	7	-7	1
		2.	-5	0	3	-8	-2	-1	-6
		3.	-7	1	2	0	5	2	9
		4.	-6	4	-4	1	0	6	3

		5.	2	-2	-1	5	3	0	9
4	$W(p)=\dfrac{b_2p^2+b_0}{a_4p^4+a_3p^3+a_2p^2+a_1p+a_0}$	1.	0	-5	4	3	7	9	1
		2.	7	-6	0	5	8	2	2
		3.	-2	-8	2	0	4	3	3
		4.	-7	-1	6	9	0	4	2
		5.	-3	7	-4	4	5	0	1
5.	$W(p)=\dfrac{b_3p^3+b_2p^2}{a_4p^4+a_3p^3+a_2p^2+a_1p+a_0}$	1.	0	-5	4	3	7	9	1
		2.	7	-6	0	5	8	2	2
		3.	-2	-8	2	0	4	3	3
		4.	-7	-1	6	9	0	4	2
		5.	-3	7	-4	4	5	0	1

Часть 3: Изучение частотных характеристик САУ (методом корневого годографа)

Постановка задачи

Имеется модель разомкнутой системы, передаточная функция которой представлена в виде произведения передаточных функций типовых звеньев систем автоматического управления БТС (САУ) ее образующих:

$$W(s)=\frac{K\cdot s^{\alpha_1}\prod_{j=1}^{\beta_1}(T_js+1)\prod_{j=1}^{\eta_1}(T_j^2s^2+2T_j\zeta s+1)}{s^{\alpha_2}\prod_{i=1}^{\beta_2}(T_is+1)\prod_{i=1}^{\eta_2}(T_i^2s^2+2T_i\zeta s+1)}.$$

Исследуя устойчивость ей соответствующей САУ в замкнутом состоянии в зависимости от коэффициента усиления, необходимо средствами MatLab выполнить:

1. Построить корневой годограф.

2. Получить коэффициент усиления $K^{кр}$, при котором система находится на границе устойчивости.
3. Вычислить критическую частоту, при которой в системе возникают незатухающие колебания (граница устойчивости).
4. Нанести на ветви корневого годографа значения полюсов замкнутой системы, соответствующие $0.5 \cdot K^{кр}$ и $0.25 \cdot K^{кр}$.
5. Привести выражение для $W_з(s)$ в виде произведения типовых звеньев. Указать значения параметров типовых звеньев.
6. Проанализировать влияние коэффициента усиления на запас устойчивости системы по амплитуде и фазе.

Для выполнения задачи ознакомимся с **краткими теоретическими сведениями.**

В ряде случаев, имеющих практическое значение, модель линейной системы автоматического управления (САУ) задается в виде структурной схемы, состоящей из типовых звеньев, математическое описание которых задано в операторной форме. Связь между входом и выходом системы задается в виде передаточной функции $W(s)$. В общем виде передаточную функцию $W(s)$ можно представить в виде:

$$W(s)=\frac{B(s)}{A(s)}, \quad (1)$$

где s – комплексная переменная, B(s) – полином степени m; A(s) – полином степени n.

Для физически реализуемых САУ $m \le n$. Коэффициенты полиномов являются действительными числами.

Применение метода корневого годографа (КГ) обусловлено фундаментальной зависимостью поведения линейной САУ от полюсов и нулей ее передаточной функции. Под полюсами подразумеваются корни полинома - знаменателя A(s), а под нулями - корни полинома числителя B(s). Полином A(s) называется характеристическим многочленом передаточной функции W(s). Положение полюсов W(s) на комплексной плоскости определяет

устойчивость САУ, а в совокупности с нулями вид импульсной переходной функции $w(t)$ и переходной функции $h(t)$. Метод корневого годографа позволяет находить полюса и нули передаточной функции замкнутой системы, располагая полюсами и нулями разомкнутой системы при изменении коэффициента усиления разомкнутой системы k. Метод корневого годографа является базовым методом для проектирования пропорционального устойчивого регулятора.

Передаточную функцию разомкнутой системы $W_p(s)$ представим в виде:

$$W_p(s)=\frac{K\cdot C\cdot\prod_{j=1}^{m}(s-s_j^0)}{\prod_{i}^{n}(s-s_j^*)}\ , \qquad (2)$$

Где s_j^0 – нули передаточной функции $W_p(s)$, , s_j^* – полюса передаточной функции $W_p(s)$, n и m – порядки знаменателя и числителя, K - коэффициент усиления разомкнутой системы, C – коэффициент представления (для канонической формы).

Передаточная функция разомкнутой системы, как правило, задается в виде отношения произведений передаточных функций стандартных (типовых) звеньев, при описании которых используются выражения трех видов:

$$Ts\ , \qquad (3)$$

$$Ts+1\ , \qquad (4)$$

$$T^2\cdot s^2+2T\cdot\varsigma\cdot s+1\ , \qquad (5)$$

здесь T постоянная времени (сек).

Если выражения (3), (4), (5) стоят в знаменателе передаточных функций звеньев (в числителе 1), то звенья называются соответственно интегрирующим, апериодическим, колебательным. Для колебательного звена - ς - безразмерный коэффициент затухания ($0<\varsigma<1$). Если выражения (3), (4), (5) стоят в числителе передаточных функций звеньев (1), то звенья называются соответственно дифференцирующим, форсирующим первого порядка, форсирующим второго порядка.

Для перехода от стандартной формы записи к формуле (2) необходимо вычислить полюса и нули соответствующих типовых звеньев:

- использующих выражение (4) – $s^{*}=-\frac{1}{T}$;

1) использующих выражение (5) – $s_{1,2}^{*(0)}=-\frac{\varsigma}{T}\pm\sqrt{(\frac{\varsigma}{T})^{2}-\frac{1}{T^{2}}}$;

Коэффициент представления С вычисляется по формуле:

$$C=\frac{\prod_{j=1}^{m}T_{j}^{0}}{\prod_{i=1}^{n}T_{i}^{*}}\ . \quad (6)$$

Для звеньев, использующих выражение (5), соответствующая постоянная времени входит в выражение (10) в квадрате.

При замыкании системы с передаточной функцией $W_{\text{р}}(s)$ единичной обратной связью передаточная функция замкнутой системы $W_{з}(s)$ принимает вид:

$$W_{ç}(s)=\frac{W_{\text{р}}(s)}{1\pm W_{\text{р}}(s)},\ (7)$$

где знак «+» соответствует отрицательной обратной связи; знак «-» соответствует положительной обратной связи.

Структурная схема системы с обратной связью приведена на рисунке 9.

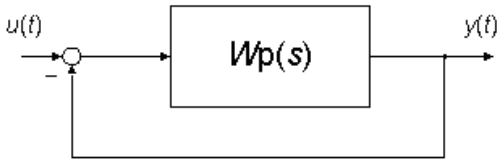


Рис. 9. Структурная схема САУ.

Из (7) следует, что нули передаточной функции замкнутой системы равны нулям передаточной функции разомкнутой системы.

Для определения полюсов замкнутой системы необходимо решить уравнение: $W_{p}(s) = -1$.

Для определения полюсов замкнутой системы (рисунок 8) необходимо решить уравнение: $W_{p}(s) = -1$. Так как $W_{p}(s)$ является

функцией комплексного переменного s, то уравнение распадается на два уравнения: – уравнение модулей $|W(s)|=1$ и уравнение аргументов.

Геометрическое место точек на комплексной плоскости, удовлетворяющее приведенным выражениям называется корневым годографом.

Таким образом, конфигурация корневого годографа не зависит от коэффициента усиления K, но каждому конкретному значению K однозначно соответствуют точки на корневом годографе.

Для определения этого соответствия достаточно воспользоваться уравнением в следующей интерпретации: $\dfrac{K \cdot C \cdot \prod_{j=1}^{m} l_j^0}{\prod_{i=1}^{n} l_i^*} = 1$, где l_j^0 – модуль (длина) вектора, проведенного из j-нуля в точку s КГ; l_i^* – модуль вектора, проведенного из i-полюса в ту же точку s.

Приведем свойства корневых годографов (для случая отрицательной обратной связи):

1. Ветви корневого годографа непрерывны и расположены на комплексной плоскости симметрично относительно действительной оси.
2. Число ветвей КГ равно порядку системы n. Ветви начинаются в n полюсах разомкнутой системы при $K = 0$. При возрастании K от 0 до бесконечности полюса замкнутой системы двигаются по ветвям КГ.
3. Отрезки действительной оси, по которым перемещаются действительные полюса замкнутой системы являются действительными ветвями корневого годографа. Эти ветви находятся в тех частях действительной оси, справа от которых расположено нечетное общее число действительных полюсов и нулей разомкнутой системы.
4. m ветвей КГ при возрастании K от 0 до бесконечности заканчиваются в m нулях $W_p(s)$, а $(n-m)$ ветвей при K, стремящемся к бесконечности, удаляются от полюсов вдоль асимптот.

5. Асимптоты в виде звезды из $(n-m)$ полупрямых выходят из точки с координатой $\sigma_a = \dfrac{\sum_{j=1}^{m} s_j^0 - \sum_{i=1}^{n} s_i^*}{n-m}$ на действительной оси под углами $\theta_a = \dfrac{2\cdot\nu+1}{n-m}\pi, \quad (\nu = 0,1,...n-m-1)$ к действительной оси.

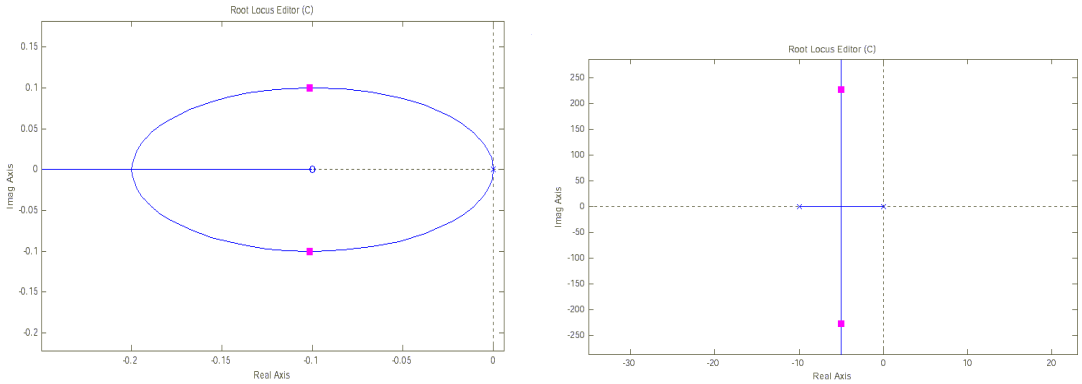
6. Угол выхода θ_i^* ветви КГ из полюса s_i^* определяется из уравнения, примененного к данному полюсу. Аналогично определяется угол входа ветви КГ в нуль s_j^0.

7. При расположении ветвей корневого годографа в левой полуплоскости s САУ устойчива. При пересечении ветвей КГ мнимой оси слева направо САУ становится неустойчивой. Пусть при $K = K^{кр}$ пересечение КГ с мнимой осью произойдет в некоторой точке i соответствующей частоте $\omega_{кр}$. Назовем это значение коэффициента усиления критическим $K^{кр}$, а величину $\omega_{кр}$ критической угловой частотой, на которой система становится неустойчивой.

Таким образом, метод КГ позволяет выбрать коэффициент усиления САУ, подобрать расположение полюсов и нулей передаточной функции корректирующих звеньев, определить параметры доминирующих полюсов САУ (ближайших к началу координат плоскости s).

В качестве примеров, приведем КГ для двух систем автоматического управления.

На рисунке 10а приведен корневой годограф САУ, передаточная функция разомкнутой системы, которой равна: $W_{p1} = \dfrac{K\cdot(s+1)}{s^2}$. Рисунок 10б иллюстрирует КГ САУ с передаточной функцией разомкнутой системы вида: $W_{p1} = \dfrac{K}{s\cdot(s+10)}$.



а *б*

Рис. 10- примеры корневых годографов.

Последовательность выполнения работы в случае использования инструментария MATLAB.

Для выполнения лабораторной работы необходимо выбрать в меню View пункт Root Locus (корневой годограф), для отображения редактора Root Locus Editor. В правом верхнем углу SISO-Design Tool можно менять тип обратной связи (кнопка «±») и структурную схему САУ. В лабораторной работе предполагается наличие отрицательной обратной связи, и структурной схемы, показанной на рисунке 8. Для загрузки данных из рабочего пространства MATLAB необходимо использовать меню «File/Import», в результате которой появляется диалог Import System Data. Необходимо, чтобы в результате импортирования данных получилась рассматриваемая схема САУ (рисунок 8). Используя Root Locus Editor и значение коэффициента усиления (здесь C – Current Compensator), выполнить задания лабораторной работы. Изменение динамических и частотных характеристик замкнутой системы при изменении *K* можно проследить, используя меню «Tools/Loop Responses».

Таким образом, **последовательность выполнения лабораторной работы (часть 3) следующая:**

1. В соответствии с заданным вариантом нарисовать структурную схему САУ.
2. Запустить систему MATLAB.
3. Создать zpk-объект, в соответствии с заданным вариантом.

4. Определить значения полюсов и нулей разомкнутой системы $W_p(s)$.
5. Запустить SISO-Design Tool и построить КГ.
6. В соответствии с теорией проанализировать расположение ветвей корневого годографа.
7. Определить условия неустойчивости замкнутой САУ. Определить $K^{кр}$ и ^{кр}.
8. Определить значения полюсов, соответствующие $0.5K^{кр}$ и $0.25K^{кр}$.
9. Проанализировать влияние удаленных полюсов и нулей на величины $K^{кр}$ и $\omega^{êð}$.
10. При K=1 привести выражение для $W_з(s)$ в виде произведения типовых звеньев. Указать значения параметров типовых звеньев.

Методический пример:

Пусть необходимо исследовать САУ с передаточной функция разомкнутой системы:

$$W(s) = \frac{(0.2s+1)}{s(0.1s+1)(0.04s^2+2\cdot 0.2\cdot 0.3s+1)}.$$

1. Создадим ZPK-объект, найдем полюса и нули разомкнутой системы:

```
>> s = zpk('s');  W = (0.2*s+1)/(s*(0.1*s+1)*(0.2^2*s^2+2*0.2*0.3*s+1))

Zero/pole/gain:
         50 (s+5)
------------------------
s (s+10) (s^2 + 3s + 25)

>> pole(W)

ans =

         0
  -10.0000
   -1.5000 + 4.7697i
   -1.5000 - 4.7697i

>> zero(W)

ans =

     -5
```

2. Запустим SISO-Design Tool, настроим параметры и импортируем ZPK-объект из рабочего пространства MATLAB (рисунок 10). В окне Root Locus Editor интерфейса SISO-Design Tool построится корневой годограф (рисунок 11).

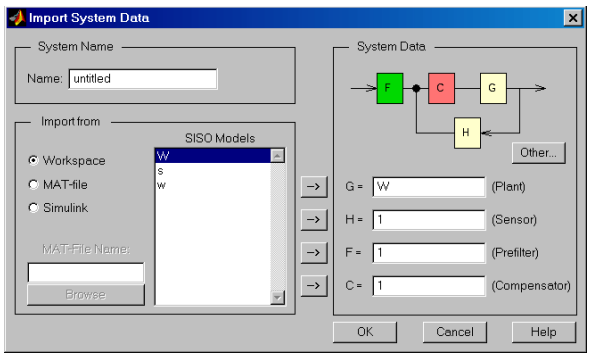


Рис. 11. Диалог импортирования данных в SISO-Design Tool.

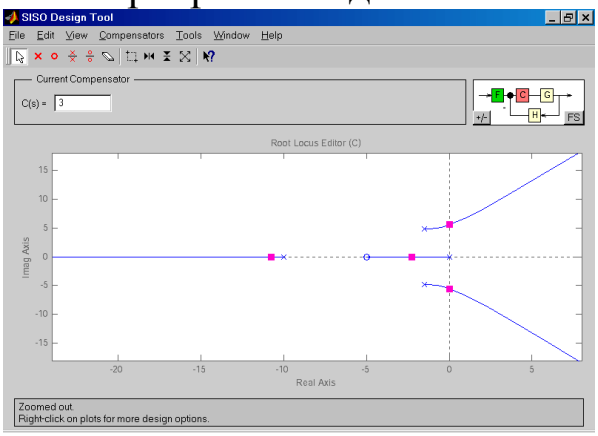


Рис. 12 - SISO-Design Tool.

3. Захватив «мышью», передвигать красным курсором по корневому годографу до пересечения ветвей с мнимой осью, определить значение $K^{кр}$ (рисунок 12). Передвижение курсора происходит также при вводе значения коэффициента усиления C в соответствующее поле ввода в верхней части GUI-интерфейса.

Для рассматриваемого случая $K^{кр}$=3. Значение $\omega_{кр}$ соответствует мнимой координате пересечения КГ мнимой оси. Просмотреть это значение можно в нижней части интерфейса или выбрав меню пункт «View/Closed-Loop Poles» (рисунок 14).

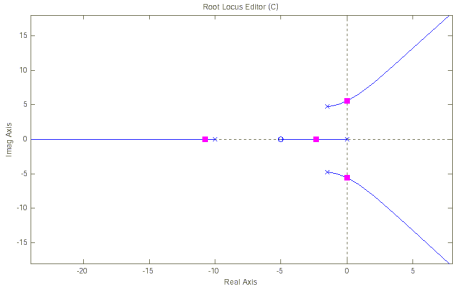


Рис. 13. Корневой годограф с нанесенным значением $K^{кр}$.

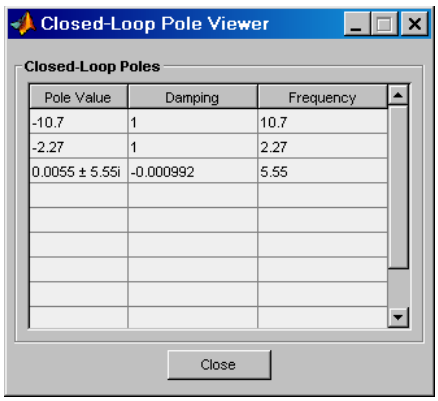
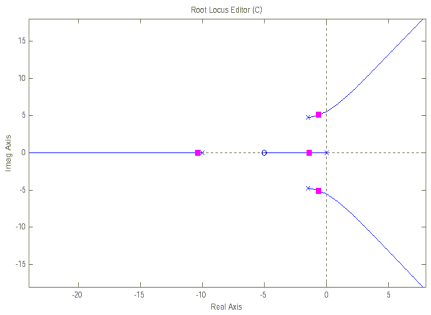
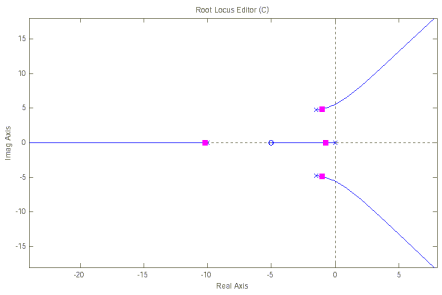


Рис. 14. Closed-Loop Poles Viewer.

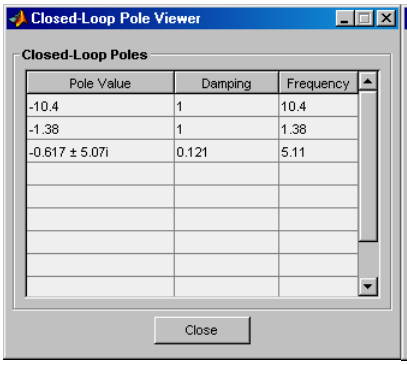
4. Зададим значения $0.5K^{кр}$ и $0.25K^{кр}$ (рисунки 15а, 15б) и определим значения полюсов (рисунки 16а , 16б).

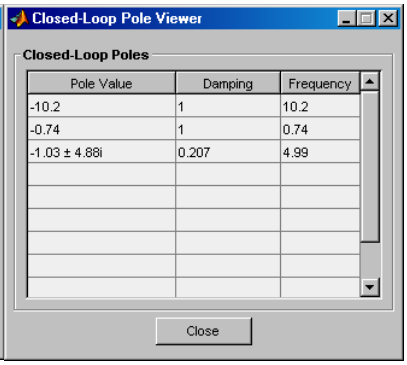




а. C= $0.5K^{кр}$ *б*. C= $0.25K^{кр}$

Рисунки 15. -Корневой годограф САУ.





а. C= $0.5K^{кр}$ *б*. C= $0.25K^{кр}$

Рисунки 16. Closed-Loop Poles Viewer.

5. Например, для значения $0.5K^{кр}$ построим вид переходной функции замкнутой системы. Для этого необходимо выбрать в меню пункт «Tools/Loop Responses/Closed-Loop Step». Результат

построения переходной функции – рисунок 17. Видно, что система устойчива.

Меняя значения *C* можно увидеть в соответствующее изменение переходной функции или других характеристик системы в динамике. При изменении *C* происходит автоматическое обновление выбранных характеристик замкнутой системы.

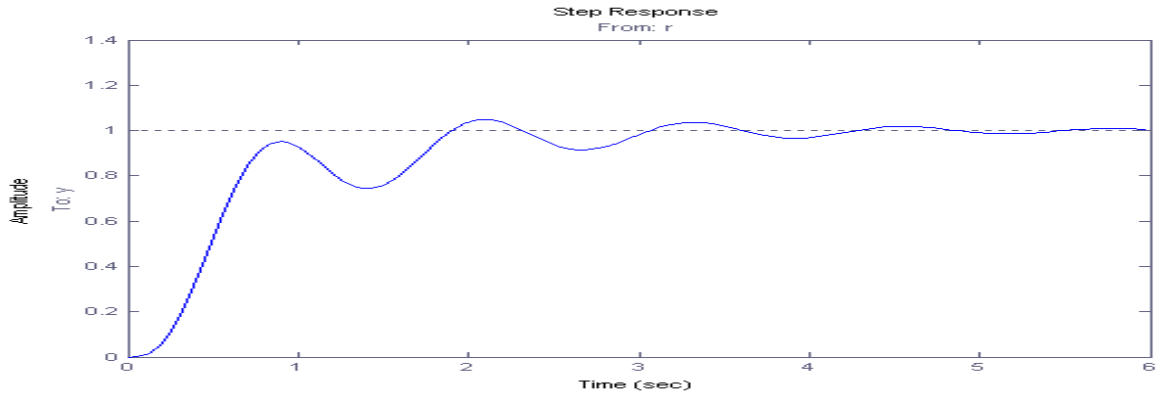


Рис. 17. Переходная функция при $C = 0.5K^{кр}$.

Таким образом, на основе использования метода корневого годографа получены области значений коэффициента усиления, в которых система автоматического управления является устойчивой.

Контрольные вопросы

1. Дать определение передаточной функции, полюсов, нулей, корневого годографа. Назвать типовые звенья САУ. Что такое отрицательная обратная связь?
2. Показать влияние расположения нуля на поведение ветвей КГ (для примера указанного преподавателем).
3. Показать на конкретном примере, что по мере удаления ветви КГ от начала координат движение ветви в зависимости от *K* замедляется.
4. Вывести зависимость для малого перемещения ветвей КГ от исходного полюса в зависимости от изменения *K*.
5. Провести анализ влияния изменения расположения полюса или нуля (по указанию преподавателя) на величины $K^{кр}$ и $\omega_{кр}$.

Варианты заданий

№	Вид передаточной функции	№№	Варианты параметров
	$W_p(s)$		Значения T_i (сек)
1.	$\frac{K(T_1 s+1)}{s(T_2 s+1)}$	1.	$T_1 = 0.5$, $T_2 = 0.1$
		2.	$T_1 = 0.1$, $T_2 = 0.01$
		3.	$T_1 = 0.1$, $T_2 = 0.9$
		4.	$T_1 = 0.01$, $T_2 = 0.1$
		5	$T_1 = 0.15$, $T_2 = 0.2$
2.	$\frac{K}{s(T^2 s^2 + 2T\zeta s+1)}$	1.	$T = 0.1$, $\varsigma = 1$
		2.	$T = 0.05$, $\varsigma = 0.707$
		3.	$T = 0.03$, $\varsigma = 0.1$
		4.	$T = 0.08$, $\varsigma = 0.5$
		5.	$T = 0.01$, $\varsigma = 0.15$
3.	$\frac{K(T_1 s+1)}{s(T_2 s+1)(T_3 s+1)(T_4 s+1)}$	1.	$T_1 = 0.03$, $T_2 = 0.5$, $T_3 = 0.1$, $T_4 = 0.05$
		2.	$T_1 = 0.05$, $T_2 = 0.4$, $T_3 = 0.08$, $T_4 = 0.033$
		3.	$T_1 = 0.2$, $T_2 = 0.45$, $T_3 = 0.1$, $T_4 = 0.05$
		4.	$T_1 = 0.5$, $T_2 = 0.25$,

			$T_3 = 0.1$, $T_4 = 0.02$
4.	$\frac{K(T_1 s+1)}{s(T_2 s+1)(T_3 s+1)(T_4^2 s^2+2T_4\zeta s+1)}$	1.	$T_1 = 0.2$, $T_2 = 0.1$, $T_3 = 0.05$, $T_4 = 0.07$, $\varsigma = 0.5$
		2.	$T_1 = 0.07$, $T_2 = 0.1$, $T_3 = 0.05$, $T_4 = 0.07$, $\varsigma = 0.5$
		3.	$T_1 = 0.3$, $T_2 = 0.1$, $T_3 = 0.05$, $T_4 = 0.07$, $\varsigma = 0.5$
		4.	$T_1 = 0.01$, $T_2 = 0.1$, $T_3 = 0.1$, $T_4 = 0.07$, $\varsigma = 0.5$
		5.	$T_1 = 0$, $T_2 = 0.1$, $T_3 = 0.1$, $T_4 = 0.07$, $\varsigma = 0.5$

Часть 4. Анализ устойчивости САУ

Устойчивость является одним из необходимых условий, обеспечивающих нормальное функционирование систем управления как в технических, так и в биотехнических системах. Поэтому чрезвычайно важно выяснить те условия, которые обеспечивают принципиальную работоспособность системы, ее устойчивость.

Признаком устойчивости системы управления является существование установившегося состояния. Если отклонение выходной координаты от заданного значения (т.е. ошибка

управления) не стремится к постоянной величине или к нулю, а возрастает или испытывает колебания, то система неустойчива.

Причинами неустойчивости могут быть инерционность элементов и большой коэффициент передачи разомкнутой системы: многократно усиленное рассогласование, возвращающееся по цепи обратной связи на вход системы, не успевает из-за запаздывания в инерционных элементах отрабатываться.

Функционирование линейной системы управления описывается уравнением динамики вида:

$$\sum_{i=0}^{n}(a_i \cdot y^{(i)}(t)) = \sum_{j=0}^{m}(b_j \cdot x^{(i)}(t)). \quad (1)$$

Ляпунов А.М. доказал следующие теоремы:

Теорема 1. Если все корни характеристического уравнения (1) имеют отрицательные действительные части, то решение уравнения асимптотически устойчиво.

Теорема 2. Если среди корней характеристического уравнения (1) имеет хотя бы один корень с положительной линейной частью, то поведение системы неустойчиво.

Передаточная функция динамической системы W(s) может быть представлена в виде дроби $W(S)=\frac{Q(s)}{P(s)}$. Устойчивость системы достигается, когда все её полюса находятся в левой полуплоскости на плоскости корней. Если W(s) получена замыканием отрицательной обратной связью разомкнутой системы с передаточной функцией $F(s)=\frac{A(s)}{B(s)}$, тогда полюса передаточной функции замкнутой системы являются нулями функции 1+F(s), а выражение 1+F(s),=0 называется *характеристическим уравнением* системы.

Отсюда вытекает основной способ оценки устойчивости – решение характеристического уравнения $\sum_{i=0}^{n}(a_i \cdot y^{(i)}(t)) = 0$. Если все корни лежат в левой полуплоскости - то система устойчива, если хотя бы у одного корня характеристического уравнения действительная часть равна 0, то система находится на границе

устойчивости. Таким образом, минимальной значение действительной части отрицательных корней (при отсутствии положительных) определяет запас устойчивости по амплитуде, а отношение модулей мнимой части корня к действительной определяет тангенс угла – запаса устойчивости по фазе. Определение корней может определяться как «вручную» (в случае небольшой степени (1)), так и с использованием инструментальных средств типа MathCard, MatLab, с помощью составления программы для определения корней методом дихатомического приближения («деления пополам»).

Назовем такой подход – **способ №1**.

Зная корни характеристического уравнения можно получить алгебраическую форму записи уравнения $y(t) = F(t)$, в которой функция в правой части определяется входным воздействием $x(t)$ (частное решение уравнения (1)). По полученной функции можно построить график функции (например, средствами MathCard, MatLab, Excel) и визуально оценить устойчивость системы управления по переходному процессу.

Назовем такой подход – **способ №2.**

Правило, позволяющее оценивать устойчивость системы (определять местоположение корней характеристического уравнения на комплексной плоскости) без непосредственного вычисления корней, называется *критерием устойчивости.* Критерии устойчивости разделяются на алгебраические и частотные.

Строгое математическое обоснование критериев устойчивости рассматривается в теории автоматического управления.

Алгебраические критерии устанавливают связь между коэффициентами характеристического уравнения и расположением его корней на комплексной плоскости. Критерий Михайлова устанавливает связь условий устойчивости с видом годографа функции комплексного переменного, представляющей собой левую часть характеристического уравнения (годограф Михайлова).

На комплексной плоскости строится годограф Михайлова – АФЧХ разомкнутой системы F(s). Согласно критерию Михайлова число оборотов кривой годографа по часовой стрелке вокруг начала

координат должно быть равно количеству нулей функции F(s) минус количество полюсов данной функции в правой полуплоскости.

Если рассматривать вместо начала координат точку на комплексной плоскости (-1, 0j), то получаем разницу между количествами нулей и полюсов в правой полуплоскости для функции 1+ F(s) (причем, данная функция 1+F(s) имеет те же полюса, что и функция F(s), а полюса разомкнутой системы являются нулями замкнутой системы). Тогда критерий Михайлова можно сформулировать следующим образом: контур в плоскости F(s) для обеспечения устойчивости должен охватывать (по часовой стрелке при изменении частоты от 0 до бесконечности) точку (-1,0j) n раз, где n=z-p (z – количество нулей, p – количество полюсов функции F(s)).

Отсюда вытекают следующие следствия:

2) **Следствия критерия Найквиста-Михайлова:**

- Если разомкнутая система с передаточной функцией F(s) устойчива, замкнутая система является устойчивой, если АФЧХ разомкнутой системы не охватывает точку (−1; j0).
- Если разомкнутая система неустойчива, то количество оборотов F(s) вокруг точки −1 должно быть равно числу полюсов F(s) в правой полуплоскости.
- Количество дополнительных охватов (больше, чем n+p вокруг точки (−1, 0j) в точности равно количеству неустойчивых полюсов замкнутой системы.

Назовем этот подход – **способ №3**.

Критерий Найквиста показывает связь условий устойчивости замкнутых систем основного типа с видом АФХ и ЛЧХ. разомкнутой системы. Рассмотрим сущность логарифмического критерия Найквиста.

Критерий устойчивости Найквиста позволяет оценивать устойчивость системы управления по логарифмическим частотным характеристикам ее разомкнутой части. Этот способ используется достаточно широко вследствие простоты построения логарифмических частотных характеристик и определения по ним

запасов устойчивости. Для оценки устойчивости системы управления сначала строят асимптотическую ЛАЧХ. Затем к ней делают поправки около тех частот, которые ограничивают положительные участки и расположены достаточно близко от сопрягающих частот (особенно от сопрягающих частот, соответствующих колебательным звеньям).

Логарифмический критерий устойчивости формулируется так: для устойчивости замкнутой САУ необходимо и достаточно, чтобы в её разомкнутой цепи запаздывание по фазе $\varphi(\omega_{ср})$ не достигало минус 180° вплоть до частоты среза ЛАЧХ при $L(\omega_{ср})==20\lg(A(\omega_{ср}))=0$ или чтобы при фазовом запаздывании $j(w\cdot k)= -180°$ величина ЛАЧХ была бы меньше нуля $L(w\cdot k)= 20\lg(A(\omega\cdot k))<0$. Анализ устойчивости при этом может производится визуально.

Назовем этот подход – **способ №4.**

В общем случае устойчивость системы определяется параметрами элементов (звеньев) системы от которых зависит передаточная функция (например: номиналы и-или электронных элементов, вязкость жидкости, температура тела и т.п.). В этом случае осуществляется построение областей устойчивости путем применения технологии D-разбиения, которое осуществляется, как правило, с помощью специализированного программного обеспечение. В случае, если исследуется влияние одного, двух или трех параметров, то область устойчивости может быть построена упомянутыми выше инструментальными средствами.

Таким образом, **последовательность выполнения части 4 практической работы №1 следующая:**

1. В соответствии с вариантом задания (рекомендуется аналогичный лабораторной работе №1) получить передаточную функцию, уравнение динамики, характеристическое уравнение системы управления.
2. Оценить устойчивость системы управления, используя инструментарии Excel, MatLab, MathCard способами 1-4.
3. Сравнить полученные результаты.
4. Сравнить трудоемкость работы. (Под показателями трудоемкости в данном случае понимаются: количество

выполненных операций и среднее затраченное время на них). Выработать рекомендации по применению определенных методов и программных инструментариев при анализе устойчивости систем управления БТС.
5. Оформить отчет.

Контрольные вопросы:
1. Какие причины лежат в основе возможной неустойчивости системы управления?
2. Как оценивается устойчивость системы управления по поведению свободной составляющей решения линейного дифференциального уравнения?
3. С какой целью выясняются условия устойчивости системы управления?
4. Что называется критерием устойчивости?
5. Какие критерии устойчивости используются в теории автоматического управления?
6. Каков физический смысл частотных характеристик?
7. Между какими функциями устанавливают связь частотные характеристики? Как они записываются и в каких координатах строятся?
8. Чем объяснить отставание по фазе выходного сигнала по отношению к входному сигналу при гармоническом характере входного сигнала?
9. Чем обусловлено увеличение фазового сдвига выходного сигнала статических систем по мере увеличения частоты входного сигнала?
10. К какому пределу стремится модуль амплитудной частотной характеристики при бесконечном увеличении частоты входной величины?
11. Как формулируется основной математический критерий устойчивости линейных систем?
12. Как по АФХ исследуемой разомкнутой системы найти критическое значение частоты и коэффициента усиления?
13. Как по ЛАЧХ исследуемой разомкнутой системы найти критическое значение частоты?

14. Какой вид имеет переходная характеристика системы, находящейся на границе устойчивости?
15. Каковы условия положения системы на границе устойчивости по критериям Михайлова, Найквиста?

Библиография

1. Березин С.Я. Основы кибернетики и управление в биологических и медицинских системах: учебное пособие / С.Я. Березин. – Старый Оскол: ТНТ, 2013. – 244 с.
2. Каганов В.И. Компьютерные вычисления в средах Excel и Mathcad. Изд-тво: Горячая Линия - Телеком, 2011. – 328 стр.
3. Наместников С.М. Основы программирования в MatLab / Сборник лекций: УлГТУ, Ульяновск. - 2011. - 55 с. URL; http://sernam.ru/lect_matlab.php
4. Моделирование в системе MatLab/ URL: http://www.knigka.info/2011/10/24/modelirovanie-v-sisteme-matlab.-chast.html
5. Основы работы в MattLab / URL: http://www.twirpx.com/file/597169/
6. Ощепков А.Ю. Системы автоматического управления. Теория, применение, моделирование в MATLAB. Из-тво: Лань. Серия: Учебники для вузов. Специальная литература, 2013. -208 с.
7. Рой О. М. Теория управления: учебное пособие. - СПб.: Питер, 2008. - 256 с.

2. РЕАКЦИЯ СИСТЕМ УПРАВЛЕНИЯ НА ТИПОВЫЕ ВХОДНЫЕ ВОЗДЕЙСТВИЯ. АНАЛИЗ ПЕРЕХОДНЫХ ПРОЦЕССОВ

Цель работы: овладение навыками исследования реакции систем автоматического управления (САУ), как представителя систем автономного управления в биомедицинских системах, на типовые входные воздействия с помощью современных компьютерных инструментариев.

Краткие теоретические сведения

Для оценки динамических свойств системы и отдельных звеньев принято исследовать их реакцию на типовые входные воздействия. Наиболее распространенными типовыми воздействиями являются ступенчатое, импульсное и гармоническое. Любой сигнал u(t), имеющий сложную форму, можно разложить на сумму типовых воздействий ui(t) и на основании принципа суперпозиции получить результирующее изменение выходной величины y(t) в виде суммы реакций системы на каждую из составляющих.

Единичная ступенька.

Особое значение в теории автоматического управления имеет ступенчатое воздействие 1(t) = 1 при t≥0, 1(t) = 0 при t<0 . Все остальные воздействия могут быть сведены к нему. Так, например, импульсный сигнал может быть представлен двумя ступенчатыми сигналами одинаковой величины противоположными по знаку, поданными один за другим через интервал времени Dt.

Преобразование Лапласа для единичной ступеньки: 1(p) = exp(-pt) dt = 1/p.

Линейно нарастающее воздействие: (t(t)=t при t≥0, t(t) = 0 при t<0) представляет собой интеграл по времени от единичной ступеньки: t(t) = 1(t) dt, 1(t) = d t(t) /dt.

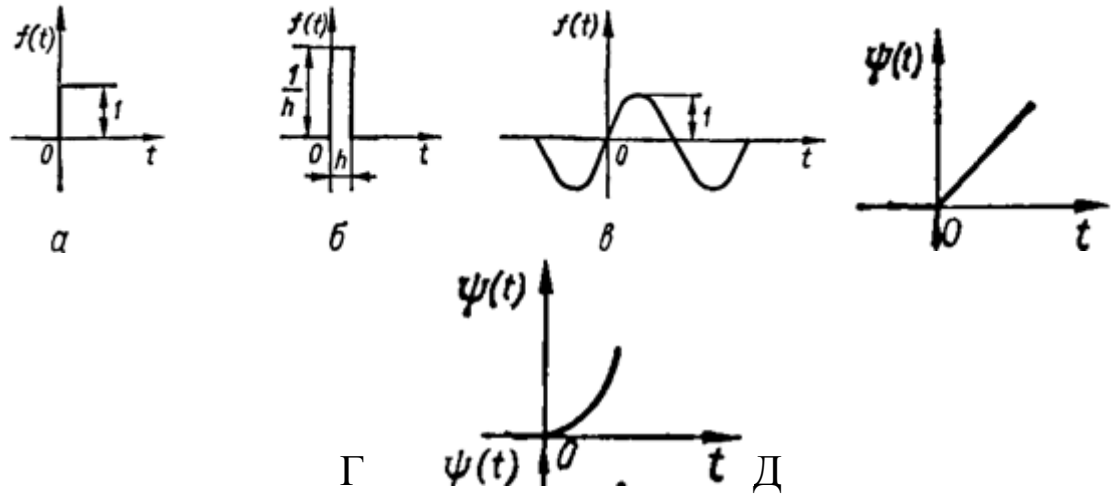
Преобразование Лапласа: t(p) = t exp(-pt), dt = 1/p2.

Экспоненциальная функция: exp(at). Преобразование Лапласа: L[exp(at)] = exp(at) exp(-pt) dt = 1/(p-a). Выражение справедливо и при любом комплексном α.

Гармонические воздействия: sin(ωt) и cos(ωt). На основе формулы Эйлера exp(jωt) = cos(ωt) + j sin(ωt) соответственно имеем cos(ωt)=Re(exp(jωt)), sin(wt)=Im(exp(jwt)). Преобразования Лапласа:
L[sin(ωt)]=L[Im(exp(jωt)]=Im(L[$e^{j\omega t}$]=Im(1/(p-jω))=
Im((p+jω)/(p2+ω2)) = Im(p/(p2+ω2)+jω/(p2+ω2)) = ω/(p2+ω2).
L[cos ωt]=Re(L(ejωt))=Re((1/(p-jω)))=Re((p+jω)/(p2+ω2))= p/(p2+ω2).

Дельта – функция: δ(t) - математическая модель очень короткого конечного воздействия большой мощности (единичный импульс). Определение δ(t)-функции даётся через интеграл свёртки с любой другой интегрируемой функцией x(t). Единичный импульс физически представляет собой очень узкий импульс, ширина которого стремится к нулю, а высота - к бесконечности, ограничивающий единичную площадь. Дельта - функция связана с единичной ступенчатой и линейно-нарастающей функцией выражением: d(t) = d1(t)/dt=d^2 t(t) /dt^2.

Основные виды типовых воздействий представлены на рисунке 1:



А — ступенчатое; б — импульсное; в — гармоническое; г — скачок скорости; д — скачок ускорения

Рис.1. Типовые возмущения систем.

Переходной функцией H(t) называется реакция звена на типовое воздействие (рис.2), т. е. переходный процесс на выходе Y(t). Как правило, под переходной функцией считают реакцию на ступенчатое воздействие.

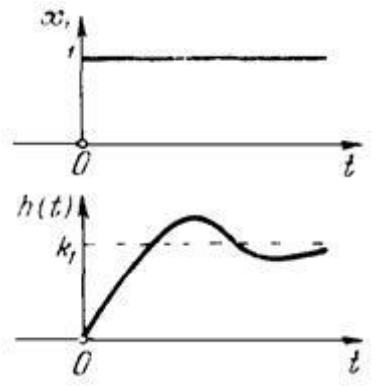


Рис.2. Входное ступенчатое воздействие x(t)=1(t) и ему соответствующая переходная функция звена h(t).

Переходная функция может быть определена экспериментально или вычислена теоретически.

Например, если на вход подается единичный скачок 1(t), то его изображение по Лапласу X(p)=L{1(t)}. Зная передаточную функцию звена, W(p)=, находим изображение выходной величины.

Переход от изображения к оригиналу может быть осуществлен по таблице операционных соответствий или по теореме разложения.

Современные компьютерные средства позволяют автоматизировать процесс построения и анализа переходных процессов, что существенным образом существенно повышает качество и результативность имитационного моделирования и вычислительного экспериментирования.

Порядок выполнения

1. Выберите передаточную функцию системы автоматического управления согласно варианта задания. (По предыдущей лабораторной работе.)
2. В среде MathLab (или любом другом инструментарии) получите реакцию системы (переходный процесс) на типовое входное воздействие (скачок, импульс). Зафиксируйте полученный результат.
3. Аналогично пункту 2, получите переходный процесс для гармонического воздействия (четные номера вариантов) и

периодического импульсного сигнала (нечетные номера вариантов) меняя частоту, амплитуду и фазу (скважность, длительность – для импульсного воздействия). Сделайте вывод о влиянии данных параметров периодического входного воздействия на выходной сигнал.

4. Повторите п.2 для замкнутой САУ с глубокой отрицательной обратной связью.

5. Повторите п.3 для замкнутой САУ с единичной положительной связью.

6. Проведите сравнительный анализ результатов п.4 и п.5

Контрольные вопросы

1. Что такое передаточная функция?
2. Свойства передаточной функции.
3. Как из дифференциального уравнения звена получить передаточную функцию?
4. Как по передаточной функции звена или системы составить дифференциальное уравнение?
5. Какие вы знаете типовые динамические звенья? Их передаточные функции и дифференциальные уравнения?
6. Какие типовые входные воздействия применяются в процессе исследования САУ?
7. В чем заключаются основные свойства дельта-функции?
8. Что такое переходная функция?
9. Что такое импульсная переходная функция?
10. Как связаны между собой $W(s)$, $h(t)$ и $w(t)$?
11. Привести примеры и временные характеристики апериодических звеньев.
12. Безынерционное (пропорциональное, усилительное) звено. Привести примеры. Временные характеристики безынерционного звена.
13. Привести примеры идеальных интегрирующих звеньев. Временные характеристики идеального интегрирующего звена.
14. Каким образом (методами, средствами) моделируются переходные процессы?

Библиография

1. Березин С.Я. Основы кибернетики и управление в биологических и медицинских системах: учебное пособие / С.Я. Березин. – Старый Оскол: ТНТ, 2013. – 244 с.
2. Каганов В.И. Компьютерные вычисления в средах Excel и Mathcad. Изд-тво: Горячая Линия - Телеком, 2011. – 328 стр.
3. Моделирование в системе MatLab/ URL: http://www.knigka.info/2011/10/24/modelirovanie-v-sisteme-matlab.-chast.html
4. Ощепков А.Ю. Системы автоматического управления. Теория, применение, моделирование в MATLAB. Из-тво: Лань. Серия: Учебники для вузов. Специальная литература, 2013. -208 с.
5. Типовые входные воздействия. Реакция САУ. Презентация. http://www.myshared.ru/slide/1303817/

3. РЕШЕНИЕ ОПТИМИЗАЦИОННЫХ ЗАДАЧ МЕТОДАМИ ЛИНЕЙНОГО И ГЕОМЕТРИЧЕСКОГО ПРОГРАММИРОВАНИЙ

Цель работы: овладение навыками использования различных методов решения задач линейного и геометрического программирований при оптимизации систем управления БТС с помощью универсальных инструментальных средств автоматизации математических расчетов.

Краткие теоретические сведения.

Линейное программирование - это направление математического программирования, изучающее методы решения экстремальных задач, которые характеризуются линейной зависимостью между переменными и линейным критерием оптимальности. Необходимым условием постановки задачи линейного программирования являются ограничения на наличие ресурсов, величину спроса, производственную мощность предприятия и другие производственные факторы.

Математическая модель любой задачи линейного программирования включает в себя:

- целевую функцию, определяющую критерий оптимальности;
- систему ограничений в форме линейных уравнений или неравенств;
- требование неотрицательности переменных.

Запишем математическую модель задачи линейного программирования следующим образом:

Определить вектор $X = (x_1, x_2, \dots x_n)$, который удовлетворяет ограничениям вида:

$$\sum_{i=1}^{n}(a_{i,j} \cdot x_{i,j}) \le b_j, (j = 1, 2, .., m), \qquad (1)$$

$$x_i \ge 0, (i = 1, 2, .., n)$$

и обеспечивает максимальное значение целевой функции

$$F(\bar{X}) = \sum_{i=1}^{n}(c_i \cdot x_i) \to \max \qquad (2)$$

Геометрическое программирование - есть метод решения одного специального класса задач нелинейного программирования, в которых критерий оптимальности и ограничения задаются в виде полиномов - выражений, представляющих собой сумму произведений степенных функций от независимых переменных. С подобными задачами иногда приходится сталкиваться в проектировании. Кроме того, некоторые задачи нелинейного программирования иногда можно свести к указанному представлению, используя аппроксимационное представление для целевых функций и ограничений.

В ряде случаев, задача геометрического программирования с помощью логорифмирования или линейного приближения может быть сведена к задаче линейного программирования.

Для задачи линейного программирования (ЛП) представленной в стандартной форме, вводится понятие *плана*.

Планом, или *допустимым решением* задачи ЛП, называется вектор $X = (x_1, \quad x_2, \quad ..., \quad x_n)$, удовлетворяющий условиям (1) и (2).

План называется *опорным*, если векторы $\boldsymbol{A}_i \quad (i = 1,2,...,n)$, являются линейно независимыми. Число положительных компонент опорного плана не может превышать n. Опорный план называется невырожденным, если он содержит n положительных компонент, в противном случае опорный план называется *вырожденным*.

Оптимальным планом или оптимальным решением задачи линейного программирования называется план, доставляющий наименьшее значение линейной функции (2).

Множество всех планов задачи линейного программирования (если они существуют) является выпуклым многогранником. Каждой угловой точке многогранника решений соответствует опорный план. Каждый опорный план определяется системой m линейно независимых векторов, содержащихся в данной системе из n векторов. Существует такая угловая точка многогранника решений, в которой линейная функция достигает своего экстремального значения.

Для отыскания оптимального плана необходимо исследовать только опорные планы. Верхняя граница количества опорных планов, содержащихся в данной задаче, определяется числом сочетаний C_n^m. При больших m и n найти оптимальный план, перебирая все опорные планы задачи, очень трудно. При числе свободных членов n-m<3, задачу ЛП можно решить графическим способом. Если число свободных членов равно и более трех, то применяются не геометрические, а вычислительные методы, наиболее распространенным из которых является симплекс-метод.

Симплекс-метод основывается па последовательном переходе от одного опорного плана задачи линейного программирования к другому, при этом значение целевой функции изменяется.

Алгоритм симплексного метода включает следующие этапы.

1. Составление первого опорного плана

Система ограничений задачи, решаемой симплексным методом, задана в виде системы неравенств. Перейдем от системы неравенств к системе уравнений путем введения неотрицательных дополнительных переменных. Векторы-столбцы при этих переменных представляют собой единичные векторы и образуют базис, а соответствующие им переменные называются базисными:

$$\sum_{i=1}^{n} a_{ij} \cdot x_i + x_{n+i} = b_j, (j = 1,2,..,m),$$

где x_{ij} - базисные переменные. x_{n+i} - свободные переменные.

Решим эту систему относительно базисных переменных:

$$x_{n+i} = b_i - \sum_{i=1}^{n} a_{ij} \cdot x_i, (j = 1,2,..,m)$$

а функцию цели перепишем в таком виде.

$$F(\overline{X}) = 0 - (-\sum_{i=1}^{n} c_i \cdot x_i)$$

Полагая, что основные переменные $x_1 = x_2 = x_3 = \ldots = x_n = 0$, получим первый опорный план $\overline{X}_1 = (0,0,...,0,b_1,b_2,...,b_i)$, $F(\overline{X}_1) = 0$, который заносим в симплексную таблицу, состоящую из коэффициентов системы ограничений и свободных членов.

Последняя строка таблицы называется индексной и заполняется коэффициентами целевой функции, взятыми с противоположными знаками.

2. Проверка плана на оптимальность

Если все коэффициенты индексной строки симплексной таблицы при решении задачи на максимум неотрицательны, то план $\overline{X}_1 = (0,0,...,0,b_1,b_2,...,b_i)$ задачи является оптимальным. Если найдется хотя бы один коэффициент индексной строки меньше нуля, то план не оптимальный и его можно улучшить. Тогда переходим к следующему этапу алгоритма.

3. Определение ведущих столбца и строки

Из отрицательных коэффициентов индексной строки выбираем наибольший по абсолютной величине, столбец, в котором находится данный коэффициент, определяем как ведущий столбец, который показывает, какая переменная на следующей итерации перейдет из свободных в базисные.

Затем элементы столбца свободных членов симплексной таблицы делим на соответствующие только положительные элементы ведущего столбца. Результаты заносим в отдельный столбец θ. Строка симплексной таблицы, соответствующая минимальному значению θ, является ведущей Она определяет переменную x_i, которая на следующей итерации выйдет из базиса и станет свободной.

Элемент симплексной таблицы, находящейся на пересечении ведущих столбца и строки, называют разрешающим.

4. Построение нового опорного плана

Переход к новому плану проводится пересчетом симплексной таблицы по методу Жордана-Гаусса. Для этого сначала заменим переменные в базисе, т.е вместо x_j в базис войдет переменная x_i,. соответствующая направляющему столбцу.

Разделим все элементы ведущей строки предыдущей симплексной таблицы на разрешающий элемент и результаты деления внесем в строку следующей симплексной таблицы, соответствующей введенной в базис переменной x_i. В результате этого на месте разрешающего элемента в следующей симплексной

таблице будем иметь 1, а в остальных клетках i, включая клетку столбца индексной строки, записываем нули.

Остальные новые элементы нового плана находятся по правилу прямоугольника:

$$НЭ = СТЭ - \frac{A \cdot B}{РЭ},$$

где *НЭ* – новый элемент, *СТЭ* - элемент старого плана, *РЭ* - разрешающий элемент, *A* и *B* - элементы старого плана, образующие прямоугольник с элементами *СТЭ* и *РЭ*.

5. Проверка нового опорного плана па оптимальность в соответствии с этапом 2 алгоритма. При решении задачи линейного программировании па минимум целевой функции, признаком оптимальности плана являются отрицательные значения всех коэффициентов индексной строки симплексной таблицы.

Если в направляющем столбце все коэффициенты $a_{ij} \leq 0$, то функция цели $F(\overline{X})$ не ограничена на множестве допустимых планов, т.е. $F(\overline{X}) > 0$ и задачу решить нельзя.

Если в столбце симплексной таблицы содержатся два или несколько одинаковых наименьших значения, то .новый опорный план будет вырожденным (одна или несколько базисных переменных станут равными нулю). Вырожденные планы могут привести к зацикливанию, т.е. многократному повторению процесса вычислений, не позволяющему завершить задачу. С целью исключения этого для выбора направляющей строки используют способ Креко, который заключается в следующем.

Делим элементы строк, имеющие одинаковые наименьшее значение θ, на предполагаемые разрешающие элементы, а результаты заносим в дополнительные строки. За ведущую строку выбирается та, в которой раньше встречается меньшее число при чтении таблицы слева направо по столбцам.

Если в оптимальный план вошла дополнительная переменная x_{n+i} , то при реализации такого плана имеются недоиспользованные ресурсы i-го вида в количестве, полученном в столбце свободных членов симплексной таблицы.

Если в индексной строке симплексной таблицы оптимального плана находится нуль, принадлежащий свободной переменной, не

вошедшей в базис, а в столбце, содержащем этот нуль, имеется хотя бы один положительный элемент, то задача имеет множество оптимальных планов. Свободную переменную, соответствующую указанному столбцу, можно внести в базис, выполнив соответствующие этапы алгоритма. В результате будет получен второй оптимальный план с другим набором базисных переменных.

Рассмотрим пример решения задачи симплексным методом.

Задача. Фармацевтический завод выпускает лекарства видов А, В и С. Плановые нормативы затрат ингредиентов на единицу лекарственного средства каждого вида, прибыль от продажи товаров на тыс. руб., а также объем ресурсов заданы в таблице 1.

Таблица 1

Виды ингредиентов	Норма ингредиентов на единицу лекарства каждого вида, кг.			Объем ресурсов b_i
	А	В	С	
Масло	0,1	0,2	0,4	1100
Кислота	0,05	0,02	0,02	120
Глюкоза	3	1	2	8000
Прибыль, тыс.руб.	3	5	4	-

Запишем математическую модель задачи.

Определить $\overline{X} = (x_1, x_2, x_3)$, который удовлетворяет условиям:

$$\begin{cases} 0{,}1x_1 + 0{,}2x_2 + 0{,}4x_3 \le 1100 \\ 0{,}05x_1 + 0{,}02x_2 + 0{,}02x_3 \le 120 \\ 3x_1 + x_2 + 2x_3 \le 8000 \\ x_1 \ge 0, x_2 \ge 0, x_3 \ge 0 \end{cases}$$

и обеспечивают максимальное значение целевой функции

$$F(X) = (3x_1 + 5x_2 + 4x_3) \to \max$$

Для построения первого опорного плана систему неравенств, переходим к системе уравнений:

$$0{,}1x_1 + 0{,}2x_2 + 0{,}4x_3 + x_4 = 1100$$
$$0{,}05x_1 + 0{,}02x_2 + 0{,}02x_3 + x_5 = 120$$
$$3x_1 + x_2 + 2x_3 + x_6 = 8000$$

В матрице этой системы уравнений $A=a_{i,j}$ имеет:

$$A = \begin{pmatrix} 0{,}1 & 0{,}2 & 0{,}4 & 1 & 0 & 0 \\ 0{,}05 & 0{,}02 & 0{,}02 & 0 & 1 & 0 \\ 3 & 1 & 2 & 0 & 0 & 1 \end{pmatrix}$$

где векторы A_4, A_5, A_6 - линейно независимы, так как определитель, составленный из компонент этих векторов, отличен от нуля.

Соответствующие этим векторам переменные x_4, x_5, x_6 будут базисными. Решим систему уравнений относительно базисных переменных.

$$x_4 = 1100 - (0{,}1x_1 + 0{,}2x_2 + 0{,}4x_3)$$
$$x_5 = 120 - (0{,}05x_1 + 0{,}02x_2 + 0{,}02x_3)$$
$$x_6 = 8000 - (3x_1 + x_2 + 2x_3)$$

Целевая функция в этом случае имеет вид:

$$F(X) = 0 - (-3x_1 - 5x_2 - 4x_3)$$

Полагая, что свободные переменные x_1=0, x_2=0, x_3 =0, получим первый опорный план X,=(*0,0,0,1100,120,8000*), $F(\overline{X}_1) = 0$, в котором базисные переменные x_4=1100, x_5=120, x_6=8000, следовательно товары не продаются и прибыль равна нулю, а ресурсы не используются.

Запишем первый опорный план в симплексную таблицу:

Таблица 2

План	Базисные переменные	Ресурсы плана	Значения коэффициентов при переменных						θ_i
			x_1	x_2	x_3	x_4	x_5	x_6	
I план	x_4	1100	0,1	**0,2**	0,4	1	0	0	5500
	x_5	120	0,05	0,02	0,02	0	1	0	6000
	x_6	8000	3	1	2	0	0	1	8000
Инд. строка	F(**X**1)	0	-3	-5	-4	0	0	0	
II план	x_2	5500	0,5	1	2	5	0	0	11000
	x_5	10	0,04	0	-0,02	-0,1	1	0	250
	x_6	2500	2,5	0	0	-5	0	1	1000
Инд. строка	F(**X**2)	27500	-0,5	0	6	25	0	0	
III план	x_2	5355	0	1	2,25	6,25	12,5	0	
	x_1	250	1	0	-0,5	-2,5	25	0	
	x_6	1875	0	0	1,25	1,25	-62,5	1	
Инд. строка	F(**X**3)	27625	0	0	5,75	23,75	12,5	0	

Первый опорный план не оптимальный, так как в индексной строке находятся отрицательные коэффициенты -3,-5,-4.

За ведущий столбец выберем столбец, соответствующий переменной x_2, так как, сравнивая по модулю, имеем:

$|-5| > \{|-3|, |-4|\}$

Рассчитываем значения θ_i по строкам как частное от деления $\frac{b_i}{a_{ij}}$ и выбираем наименьшее:

$$\min\left(\frac{b_i}{a_{ij}}\right) = \min\left(\frac{1100}{0,2}; \frac{120}{0,02}; \frac{8000}{1}\right) = 5500.$$

Следовательно, первая строка является ведущей. Элемент ***0,2*** находится на пересечении ведущего столбца и ведущей строки и выделен.

Формируем следующую симплексную таблицу. Вместо переменной x_4 в план II войдет переменная x_2. Строка, соответствующая переменной x_2 в плане II, получена в результате деления всех элементов строки x_4 плана I на разрешающий элемент

РЭ=0,2. Па месте разрешающего элемента в плане II получаем *1*. В остальных клетках столбца x_2 плана II записываем нули.

Таким образом, в новом плане II заполнены строки x_2 и столбец x_4. Все остальные элементы нового плана II, включая элементы индексной строки, определяется по правилу прямоугольника. Для этого выбираем из старого плана четыре числа, которые расположены в вершинах прямоугольника и всегда включают разрешающий элемент ***РЭ=0,2***. Во второй вершине по диагонали находится старое значение элемента, например, значение целевой функции $F(K_i) = 0 = СЭ$, которое указывает на месторасположение нового НЭ в новом плане II. Третий элемент *A=1100* и четвертый элемент *B=-5* завершают построение прямоугольника в недостающих двух вершинах и расположены по другой диагонали. Значение нового элемента в плане II находится из выражения:

$$НЭ = СТЭ - \frac{A \cdot B}{РЭ} = 0 - \frac{1100 \cdot (-5)}{0,2} = 27500$$

Элементы строки определяются аналогично:

$$120 - \frac{1100 \cdot 0,02}{0,2} = 10$$

$$0,05 - \frac{0,1 \cdot 0,02}{0,2} = 0,04$$

$$0,02 - \frac{0,4 \cdot 0,02}{0,2} = -0,02$$

$$0 - \frac{0,02 \cdot 1}{0,2} = -0,1$$

Все элементы, расположенные на пересечении строк и столбцов, соответствующих одноименным базисным элементам равны *1*, остальные элементы столбца и базисах векторов, включая индексную строку, равны *0*. Аналогично проводятся расчеты по всем строкам таблицы, включая индексную.

Выполняя последовательно все этапы алгоритма, формируем план II.

На третьей итерации таблицы 3 получаем план III, который является оптимальным, так как все коэффициенты и индексной строке больше или равны нулю. Оптимальный план можно записать так:

$$\overline{X} = (250,5375,0,0,0,1875) \quad F(\overline{X}) = 27625 тыс.руб.$$

Следовательно, необходимо продавать лекарства вида А в количестве *250* ед., а вида В – в количестве *5375* ед. При этом завод по производству лекарственных средств получает максимальную прибыль в размере *27625* тыс. руб. Лекарство вида С не реализуются.

В оптимальном плане среди базисных переменных находится дополнительная переменная x_6. Это указывает, что ресурсы третьего вида (сахар) недоиспользована на *1875* м, так как переменная x_6 была введена в первое ограничение задачи, характеризующее собой использование этого ресурса.

В индексной строке III плана в столбцах переменных x_3, x_4, x_5, не вошедших в состав базисных, получены ненулевые элементы, поэтому оптимальный план задачи линейного программирования является единственным.

Решение задач линейного программирования средствами MathCAD

Программный пакет для автоматизации математических расчетов MathCAD позволяет решить задачу линейного программирования при помощи решающего блока и функций *Minimize (Maximize)*.

Функции *Minimize (Maximize)* имеют синтаксис:

Minimize(*f, var1, var2, ...*) - Функция возвращает такие значения переменных *var1, var2, ...,* которые отыскивают минимальное значение функции *f* и удовлетворяют системе ограничений, записанных внутри решающего блока.

Maximize(*f, var1, var2, ...*) - Функция возвращает такие значения переменных *var1, var2, ...,* которые отыскивают максимальное значение функции *f* и удовлетворяют системе ограничений, записанных внутри решающего блока.

Алгоритм решения задачи линейного программирования средствами MathCAD следующий:

1. Вводят исходные данные задачи (в нашем случае это переменные *n*- количество ограничений, *m* – количество переменных, *c, a, b*)
2. Вводят линейную целевую функцию $f(x_1..x_n)$.
3. Задают начальные значения переменным задачи (в нашем случае это переменные $x_1..x_n$).
4. Формируют решающий блок со служебным словом *Given*, внутри которого вводят ограничения задачи в матричной форме (в случае небольшого числа переменных можно ввести ограничения в естественной форме).
5. Завершают решающий блок обращением к функции *Minimize (Maximize)*.

***Пример*:** Найти максимальное значение функции

$L(x) = 8x_1 + 2x_2 - 5x_3 \rightarrow max$

при заданных ограничениях

$6 \le -x_1 + 2x_2 + x_3$

$3 \le x_1 - 2x_2 + 2x_3$

$2 \ge 2x_1 + x_2 - x_3$

$x_1, x_2, x_3 \ge 0$

Решение

1. Вводят исходные данные задачи в матричной форме.

$$m := 3$$

$$n := 3$$

$$c := \begin{pmatrix} 8 \\ 2 \\ -5 \end{pmatrix} \qquad a := \begin{pmatrix} -1 & 2 & 1 \\ 1 & -2 & 2 \\ -2 & -1 & 1 \end{pmatrix} \qquad b := \begin{pmatrix} 6 \\ 3 \\ -2 \end{pmatrix}$$

2. Вводят линейную целевую функцию.

$$f(x) := \sum_{j=1}^{n} c_j \cdot x_j$$

3. Задают начальные значения переменным задачи.

$$x_n := 0$$

4. Вводят ограничения задачи в матричной форме.

Given

$a \cdot x \le b$

$x \ge 0$

5. Определяют оптимальное решение задачи с помощью встроенной функции *Maximize* (в случае поиска максимума функции) или *Minimize* (в случае поиска минимума функции).

$x := \mathrm{Maximize}(f, x)$

$$x = \begin{pmatrix} 1.4 \\ 2.2 \\ 3 \end{pmatrix}$$

$f(x) = 0.6$

Пример использования инструментария Excel для решения задач линейного программирования.

Задача о назначениях

В конкурсе на занятие пяти вакансий (V1, V2, V3, V4, V5) участвуют семь претендентов (P1, P2, P3, P4, P5, P6, P7). Результаты тестирования каждого претендента, на соответствующие вакансии, даны в виде матрицы - C (тестирование производилось по десятибалльной системе).

Требуется определить, какого претендента и на какую вакансию следует назначить таким образом, чтобы сумма баллов («вес», «профессионализм») всех претендентов оказалась максимальной.

	V1	V2	V3	V4	V5
P1	7	5	7	6	7
P2	6	4	8	4	9
P3	8	6	4	3	8
P4	7	7	8	5	7
P5	5	9	7	9	5
P6	6	8	6	4	7
P7	7	7	8	6	4

I. Математическая модель задачи.

1) Переменные задачи. Введем переменные x_{ij} принимающие два значения: $x_{ij}=0$, если *i*-й претендент (P_i) не принимается на *j*-ю вакансию (V_j). $x_{ij}=1$, если *i*-й претендент (P_i) принимается на вакансию (V_j). i=1,2,...7; j=1,2,...5.

2) Ограничения на переменные задачи.

Очевидно, что все переменные задачи неотрицательные и целые числа: $x_{ij} \geq 0$ и x_{ij} - целые.

Кроме того, так как каждый претендент может занять только одну вакансию и все вакансии должны быть заняты, должны удовлетворяться следующие ограничения:

$\sum_{i=1}^{7} x_{ij} = 1$, j=1,2,...7 , $\sum_{j=1}^{5} x_{ij} = 1$, i=1,2,...5 ,

То есть, в матрице (x_{ij}) суммы элементов по каждой строке и суммы элементов по каждому столбцу должны быть равны единицам. Это условие означает, что выбор претендентов должен быть таким, чтобы в матрице (x_{ij}), представляющей решение задачи, было бы по одной единице в каждой строке и по одной единице в каждом столбце, остальные элементы матрицы должны равняться нулю.

3) Целевая функция в задаче о назначениях.

Необходимо выбрать претендентов так, чтобы суммарное число очков, набранное ими было бы максимальным. Суммарное число набранных очков вычисляется по формуле:

$Z = \sum_{i-1}^{7} \sum_{j=1}^{5} x_{ij} c_{ij}$;

$Z=c_{11}x_{11}+c_{12}x_{12}+...+c_{75}x_{75}=7x_{11}+5x_{12}+...+4x_{75}$;

Окончательная математическая модель задачи записывается как:

требуется найти max $Z = \sum_{i-1}^{7} \sum_{j=1}^{5} x_{ij} c_{ij}$ при ограничениях:

$x_{i,j} \geq 0$ и $x_{i,j}$ - целые числа, i=1,2,...7; j=1,2,...5;

$\sum_{i=1}^{7} x_{i,j} = 1$, j=1,2,...7 ;

$\sum_{j=1}^{5} x_{i,j} = 1$, i=1,2,...5 .

Таким образом, задача о назначениях есть частный случай задачи линейного программирования.

Решение задачи в процедуре «Поиск решения» EXCEL

1) *Ввод данных*. Переносим данные задачи в EXCEL, при этом нужно ввести 2 столбца (6-ой и 7-ой) с нулевыми значениями для сбалансирования задачи. Результаты заполнения таблицы EXCEL можно увидеть на рисунке:

	A	B	C	D	E	F	G	H	I	J
1				Баллы						
2	Претенденты			Вакансии						Сумма балов
3		1	2	3	4	5	6	7		0
4	1	7	5	7	6	7	0	0		
5	2	6	4	8	4	9	0	0		
6	3	8	6	4	3	8	0	0		
7	4	7	7	8	5	7	0	0		
8	5	5	9	7	9	5	0	0		
9	6	6	8	6	4	7	0	0		
10	7	7	7	8	6	4	0	0		
11										
12										
13		1	2	3	4	5	6	7		
14	1								0	
15	2								0	
16	3								0	
17	4								0	
18	5								0	
19	6								0	
20	7								0	
21		0	0	0	0	0	0	0		

В ячейках B4:F10 введены результаты тестирования претендентов, а в ячейках G4:H10 введены нули, что соответствует фиктивным вакансиям.

Ячейки B14 : F20 являются изменяемыми ячейками для нашей процедуры.

В ячейках B21:H21 находятся суммы значений соответствующих столбцов изменяемых ячеек. Так в ячейке B21 находится сумма ячеек B14:B20. Аналогично в ячейках: в C21 находится сумма ячеек C14:C20;

в D21 находится сумма ячеек D14:D20;

в E21 находится сумма ячеек E14:E20;

в F21 находится сумма ячеек F14:F20.

в G21 находится сумма ячеек G14:G20;

в H21 находится сумма ячеек H14:H20.

В ячейках I14 : I20 находятся суммы значений соответствующих строк изменяемых ячеек. Так в ячейке I14 находится сумма ячеек B14:H14. Аналогично в ячейках: в I15 находится сумма ячеек B15:H15;

в I16 находится сумма ячеек B16:H16;

в I17 находится сумма ячеек B17:H17;

в I18 находится сумма ячеек B18:H18;

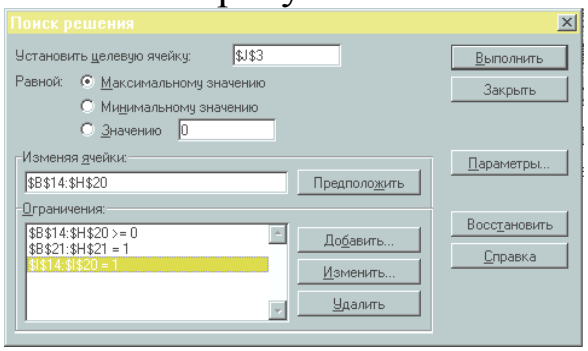
в I19 находится сумма ячеек B19:H19;

в I20 находится сумма ячеек B20:H20.

Целевая функция заносится в ячейку J3 и вычисляется по формуле «СУММПРОИЗВ(B4:H10;B14:H20)».

2) *Заполнение окна процедуры «Поиск решения»:* целевая функция: J3; значение целевой функции: max; изменяемые ячейки: B14:H20; ограничения задачи: B21:H21=1 и I14:I20=1(все свободные рабочие места должны быть заняты); B14:F20≥0 (изменяемые ячейки должны иметь положительные значения).

В окне «Параметры» установить «Линейная модель», что соответствует решению задачи симплекс-методом. Результаты заполнения окна показаны на рисунке:



3) Выполнив процедуру «Поиск решения» мы получили в первоначальной таблице следующие результаты:

	A	B	C	D	E	F	G	H	I	J
1				Баллы						
2	Претенденты			Вакансии						Сумма балов
3		1	2	3	4	5	6	7		42
4	1	7	5	7	6	7	0	0		
5	2	6	4	8	4	9	0	0		
6	3	8	6	4	3	8	0	0		
7	4	7	7	8	5	7	0	0		
8	5	5	9	7	9	5	0	0		
9	6	6	8	6	4	7	0	0		
10	7	7	7	8	6	4	0	0		
11										
12										
13		1	2	3	4	5	6	7		
14	1	0	0	0	0	0	1	0	1	
15	2	0	0	0	0	1	0	0	1	
16	3	1	0	0	0	0	0	0	1	
17	4	0	0	1	0	0	0	0	1	
18	5	0	0	0	1	0	0	0	1	
19	6	0	1	0	0	0	0	0	1	
20	7	0	0	0	0	0	0	1	1	
21		1	1	1	1	1	1	1		

Порядок выполнения работы.

1. Решить задачу ЛП симплекс-методом согласно номеру своего варианта (допускается использование компьютерных технологий).
2. Проверить решение задачи в MathCAD и Excel.
4. Решить задачу геометрического программирования в условиях варианта п.1 заменив линейные аргумента на степенные и применив инструментарий п.2. Степень аргумента выбирается случайным образом из интервала [1-1/n; 1+1/n], n – половина от номера варианта, использовать генератор случайных чисел в Excel.
5. Оформить отчет

Приложение: варианты заданий:

1. $L(x) = 2x_1 - 5x_2 \to min$
$3 >= x_1 - x_2$
$4 >= 2x_1 + x_2$
$x_1, x_2 >= 0$

2. $L(x) = 8x_1 + 2x_2 \to max$
$16 \geq x_1 + x_2$
$20 \geq 4x_1 - 2x_2$
$x_1, x_2 >= 0$

3. $L(x) = x_1 + 2x_2 \to min$
$3 \leq 2x_1 + x_2$
$1 \geq 2x_1 - 7x_2$
$6 \leq 2x_1 + 3x_2$
$x_1, x_2 \geq 0$

4. $L(x) = 9x_1 + 8x_2 - 3x_3 \to max$
$5 \leq -x_1 + 2x_2 + x_3$
$6 \leq x_1 - 2x_2 + 2x_3$
$7 \geq 2x_1 + x_2 - x_3$
$x_1, x_2, x_3 \geq 0$

5. $L(x) = 4x_1 + 3x_2 + 6x_3 \to max$
$40 \geq 2x_1 + x_2 + x_3$
$100 \geq 4x_1 + 9x_3$
$30 \geq 3x_2 + 5x_3$
$60 \geq 2x_1 + x_2 + 5x_3$
$x_1, x_2, x_3 \geq 0$

6. $L(x) = 20x_1 + 10x_2 + 9x_3 + 10x_4 \to max$
$60 \geq 7x_1 + 7x_2 + 3x_3$
$35 \geq 3x_2 + x_3 + 3x_4$
$75 \geq 8x_1 + 4x_3 + x_4$
$50 \geq 5x_1 + 2x_2 + 3x_4$
$x_1, x_2, x_3, x_4 \geq 0$

7. $L(x) = x_1 + 2x_2 - 3x_3 + 4x_4 \to max$
$100 = x_1 - x_2 + 7x_3 + x_4$
$800 = 2x_1 + 3x_2 - x_3 + 10x_4$
$x_1, x_2, x_3, x_4 \geq 0$

8. $L(x) = x_1 + x_2 \to max$
$12 \geq 3x_1 + 4x_2$
$8 \geq 2x_1 - x_2$
$x_1, x_2 \geq 0$

9. $L(x) = 3x_1 + 2x_2 \to max$
$8 \geq 4x_1 + 3x_2$
$3 \geq 2x_1 + 0.5x_2$
$x_1, x_2 \geq 0$

10. $L(x) = x_1 + x_2 \to max$
$10 \geq 3x_1 + 2x_2$
$2 \leq x_1 \leq 5$
$0 \leq x_2 \leq 3$

11. $L(x) = -x_1 + 2x_2 \to max$
$23 \geq 5x_1 + 4x_2$
$20 \geq 3x_1 + 2x_2$
$6 \geq 3x_1 - x_2$
$x_1, x_2 \geq 0$
$x_1, x_2 \geq 0$

12. $L(x) = 6x_1 - 5x_2 \to max$
$4 \leq x_1 + x_2$
$24 \geq x_1 + x_2$
$12 \geq -x_1 + x_2$
$12 \geq x_1 - x_2$

13. $L(x) = 3x_1 + 4x_2 \to max$
$23 \geq x_1 + x_2$
$17 \geq -x_1 + x_2$
$13 \leq x_1 - 3x_2$
$x_1, x_2 \geq 0$

14. $L(x) = -2x_1 + 4x_2 + 5x_3 \to max$
$76 \geq 3x_1 + 5x_2 + 7x_3$
$x_1 \geq 0$
$0 \leq x_2 \leq 7$
$1 \leq x_3 \leq 5$

15. $L(x) = 2x_1 - x_2 + x_3 \to max$
$6 \leq x_1 + x_2 + x_3$
$2 \geq 2x_1 - x_2 + x_3$
$x_1, x_2, x_3 \geq 0$
$x_1, x_2 \geq 0$

16. $L(x) = x_1 + 4x_2 \to max$
$6 \leq x_1 - 3x_2$
$10 \geq x_1 + x_2$
$9 \leq 3x_1 + x_2$

17. $L(x) = -2x_1 + x_2 - 3x_3 \to max$
$8 \geq 3x_1 - x_2$
$1 \geq -x_1 + x_2 + 4x_3$
$6 \geq 2x_1 + x_2 - 3x_3$
$x_1, x_2, x_3 \geq 0$
$x_1, x_2 \geq 0$

18. $L(x) = -x_1 + 2x_2 \to min$
$6 \leq x_1 + 3x_2$
$1 \geq -x_1 + 2x_2$
$5 \geq x_1 + x_2$
$6 \leq 3x_1 - x_2$

19. $L(x) = 7x_1 + 4x_2 - 3x_3 + 2x_4 \to max$
$45 \geq 2x_1 + 3x_2 - 4x_3 + 5x_4$
$68 \geq x_1 + x_2 + 2x_3 + x_4$
$x_1, x_2, x_3, x_4 \geq 0$

20. $L(x) = x_1 + x_2 + x_3 + x_4 \to min$
$7 \leq 3x_1 + 2x_2 + x_4$
$3 \leq x_1 + 5x_2 + 6x_3 + 11x_4$
$x_1, x_2, x_3, x_4 \geq 0$

21. $L(x) = 2x_1 + 3x_2 + 4x_3 + 7x_4 \to min$
$40 \geq 3x_1 + 4x_2 + 5x_3 + 5x_4$
$60 \geq x_1 + x_2 + x_3 + x_4$
$x_1 \geq 1$
$x_2 \geq 0$
$x_3 \geq 1.5$

22. $L(x) = 2x_1 - 6x_2 - 4x_3 \to min$
$16 \geq -2x_1 + 6x_3$
$2 \geq 2x_1 + 8x_2 - 2x_3$
$12 \geq 2x_1 - 6x_2 + 4x_3$
$x_1 \geq 2$
$x_2 \geq 0,\ x_3 \geq 0.5$

Контрольные вопросы:

1. Что такое «линейное программирование»?
2. Что такое «геометрическое программирование»?
3. Как формулируются прямая и обратная задачи ЛП?.
4. Что такое симплекс-план?
5. В каких случаях план является вырожденным, оптимальным?
6. Что надо делать для отыскания оптимального плана?
7. В чем заключается симплекс-метод?

8. Какие задачи решаются с помощью линейного программирования?
9. Какие задачи решаются с помощью геометрического программирования?
10. В каких случаях применяется ЛП и ГП в АСУ здравоохранения?
11. В каких случаях применяется ЛП и ГП для оптимизации работы лечебно-профилактического учреждения?

Библиография

1. Давыдов Е. Г. Элементы исследования операций:[Текст]: учебное пособие / Евгений Георгиевич Давыдов. - М.: КноРус, 2010. - 160 с.
2. Линейное программирование. Симплекс-метод [Электронный ресурс]: индивидуальные задания и методические указания к модулю 18 / Юго-западный государственный университет, Кафедра высшей математики; сост.: Е. В. Журавлева, Н. А. Епишева. - Курск: ЮЗГУ, 2010. - 40с.
3. Нелинейное программирование Геометрический способ решения ЗНЛП Метод неопределенных множителей Лагранжа. – презентация / http://www.myshared.ru/slide/1258077/
4. Основная задача линейного программирования Геометрическая интерпретация. – презентация. / http://www.myshared.ru/slide/517780/
5. Пантелеев, А. В. Методы оптимизации в примерах и задачах [Текст] : учебное пособие / А. В. Пантелеев, Т. А. Летова. - 2-е изд., испр. - М. : Высшая школа, 2005. - 544 с.
6. Решение задач линейного программирования в Excel [Электронный ресурс]:методические указания по выполнению лабораторной работы / Курский государственный технический университет, Кафедра программного обеспечения вычислительной техники ; сост. А. В. Малышев. - Курск: КурскГТУ, 2009. - 19 с.
7. Ржевский, С. В. Исследование операций [Текст]: учебное пособие /С. В. Ржевский. - Санкт-Петербург: Лань, 2013. - 480 с.

4. ПРИМЕНЕНИЕ ДИНАМИЧЕСКОГО ПРОГРАММИРОВАНИЯ ПРИ ОПТИМИЗАЦИИ САУ БТС

Цель работы: овладение навыками применения динамического программирования для оптимизации систем управления с применением вычислительных средств.

Краткие теоретические сведения.

Применение классических методов математического анализа в системах управления с целью оптимизации процесса регулирования, в которых присутствуют живые, открытые подсистемы зачастую оказываются недостаточно эффективными. При этом, первоначально поставленная задача отыскания экстремума приводит к необходимости предварительного разрешения подзадач, которые оказываются сложнее исходной.

Между тем, решение подобных задач значительно упрощается, если решать их поэтапно, - т.е. реализуя метод *динамического программирования*.

Сущность метода состоит в том, что процедура поиска точек экстремума целевой функции многих переменных заменяется последовательным нахождением точек экстремума функции одного или небольшого числа переменных.

Т.е., *динамическое программирование* представляет собой поэтапное планирование многошагового процесса, при котором управление разбивается на ряд последовательных этапов, соответствующих различным моментам времени и-или разным координатам.

В задачах, решаемых методом динамического программирования, значение целевой функции оптимизируемого критерия, достигнутое в течении всего процесса, получают суммированием частных значений того же критерия, полученных на отдельных шагах. В этом случае, критерий называется аддитивным. Если критерий не является таковым, то применение рядов и полиномов или иных математических преобразований способно привести его к таковому с необходимой точностью.

В теории динамического программирования рассматриваются процессы, зависящие только от текущего состояния, оптимальная стратегия формулируется следующим образом.

Принцип оптимальности: Оптимальная стратегия обладает тем свойством, что, каковы бы не были первоначальные состояния системы и решения, последующее решение должно определять оптимальную стратегию относительно состояния, полученного в результате начального решения.

Под влиянием управлений $x_1, x_2, ..., x_m$ система переходит из начального состояния S_0 в конечное состояние $S_{êî í}$. За m шагов получаем целевую функцию-выигрыш: $f(x) = \sum_{i=1}^{m} (f_i(S_{i-1}, x_i))$ ($f_i(S_{i-1}, x_i)$ - выигрыш на i-ом шаге, зависящий от исходного состояния системы S_{i-1} и выбранного управления x_i).

Принцип оптимальности позволяет заключить, что при любом начальном управлении $f(x) = f_1(S_0, x_1) + \sum_{i=2}^{m} f_i(S_{i-1}, x_i)$. Поскольку данное соотношение справедливо для любых начальных решений, то, чтобы найти максимальное значение критерия оптимальности $f_{\max}(S_0)$ необходимо найти максимум по x_1 значения $f(x)$:
$f_{\max}(S_0) = \max_x \{ f_1(S_0, x_1) + f_{\max}(S_{m-1}, x_1)) \}$, где $f_{\max}(S_{m-1}, x_1)$ - оптимальный выигрыш на всех последующих шагах.

В динамическом программировании, планируя многоэтапную процедуру, управление на каждом шаге, кроме последнего, выбирается с учетом будущего, а последний шаг планируется таким образом, что бы он привел к максимуму целевой функции. Для этого делаются различные предположения о том, чем закончится предпоследний шаг, и для каждого допущения выбирают управление на последнем шаге и запоминают его до конца решения задачи. Такое оптимальное управление называют условным.

Поскольку невозможно запомнить все значения функции (в противном случае – задача получения оптимального решения сводится к переборной), то запоминается определенный

дискретный набор ее значений и применяют принцип рекуррентности в вычислениях.

Типовой алгоритм решения задач методом динамического программирования:

1. Описать строение оптимальных решений.
2. Выписать рекуррентное соотношение, связывающие оптимальные значения параметра для подзадач.
3. Двигаясь снизу вверх, вычислить оптимальное значение параметра для подзадач.
4. Пользуясь полученной информацией, построить оптимальное решение.

Для решения задач оптимизации существует специальная теория, большая заслуга в ее создании принадлежит Р. Беллману. Различают следующие виды динамического программирования (ДП):

Нисходящее ДП: задача разбивается на подзадачи меньшего размера, они решаются и затем комбинируются для решения исходной задачи. Используется запоминание для решений часто встречающихся подзадач

Восходящее ДП: Все подзадачи, которые впоследствии понадобятся для решения исходной задачи, просчитываются заранее и затем используются для построения решения исходной задачи. Этот способ лучше нисходящего ДП в смысле размера необходимого стека и количества вызова функций, но иногда бывает нелегко заранее выяснить - решение каких подзадач потребуется в дальнейшем.

Метод динамического программирования работоспособен, если формальная интерпретация реальной задачи позволяет выполнить следующие условия:

1. Рассматриваемая задача может быть представлена как N—шаговый процесс, описываемый соотношением:

$X = f(X\ U\ ,\ n)$, где n – номер одного из множества возможных состояний системы, в которое она переходит по завершении n—ого шага; X – вектор состояния системы, принадлежащий

упомянутому *n*—ному множеству; *U*– управление, выработанное на шаге *n* (шаговое управление), переводящее систему из возможного её состояния в *n*—ном множестве в одно из состояний (*n + 1*)-го множества.

2. Структура задачи не должна изменяться при изменении расчетного количества шагов *N*.

3. Размерность пространства параметров, которыми описывается состояние системы, не должна изменяться в зависимости от количества шагов *N*.

4. Выбор управления на любом из шагов не должен отрицать выбора управления на предыдущих шагах. Иными словами, оптимальный выбор управления в любом из возможных состояний должен определяться параметрами рассматриваемого состояния, а не параметрами процесса, в ходе которого система пришла в рассматриваемое состояние.

Чисто формально, если одному состоянию соответствуют разные предыстории его возникновения, влияющие на последующий выбор оптимального управления, то метод позволяет включить описания предысторий в вектор состояния, что ведёт к увеличению размерности вектора состояния системы. После этой операции то, что до неё описывалось как одно состояние, становится множеством состояний, отличающихся одно от других компонентами вектора состояния, описывающими предысторию процесса.

Рассмотрим рисунки 1-3. На рисунке 1 показаны начальное состояние системы – «0» и множества её возможных последующих состояний – «1», «2», «3», а также возможные переходы из каждого возможного состояния в другие возможные состояния. Каждому переходу-шагу соответствует свой шаговый выигрыш, а в завершающем процесс третьем множестве – каждому из состояний системы придана его оценка, помещенная в прямоугольнике.

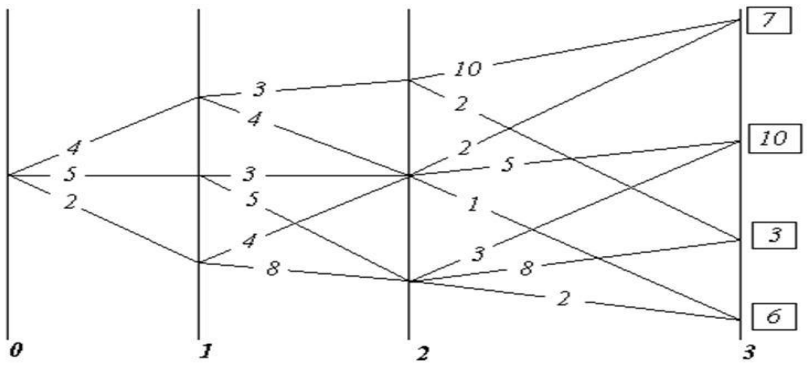


Рисунок 1 – «существо» метода динамического программирования. Матрица возможностей.

Если выбирать оптимальное управление на первом шаге, то необходимо предвидеть все его последствия на последующих шагах. Поэтому описание алгоритма метода динамического программирования часто начинают с описания выбора управления на последнем шаге, ведущем в одно из завершающих процесс состояний. При этом ссылаются на «педагогическую практику», которая свидетельствует, что аргументация при описании алгоритма от завершающего состояния к начальному состоянию легче воспринимается, поскольку опирается на *как бы уже сложившиеся* к началу рассматриваемого шага условия, в то время как возможные завершения процесса также определены.

На рисунке 2 показаны возможные переходы в завершающее множество состояний «3» из каждого возможного состояния в ему предшествующем множестве состояний «2», будто бы весь предшествующий путь уже пройден и осталось последним выбором оптимального шагового управления завершить весь процесс. При этом для каждого из состояний во множестве «2» определяются ***все*** *полные выигрыши как сумма = «оценка перехода» + «оценка завершающего состояния».* Во множестве «2» из полученных для каждого из состояний, в нём возможных полных выигрышей,

определяется и *запоминается* максимальный полный выигрыш и соответствующий ему переход (фрагмент траектории). Максимальный полный выигрыш для каждого из состояний во множестве «2» взят в прямоугольную рамку, а соответствующий ему переход отмечен стрелкой. Таких оптимальных переходов из одного состояния в другие, которым соответствует одно и то же значение полного выигрыша, в принципе может оказаться и несколько. В этом случае все они в методе неразличимы и эквивалентны один другому в смысле построенного критерия оптимальности выбора траектории в пространстве параметров, которыми описывается система.

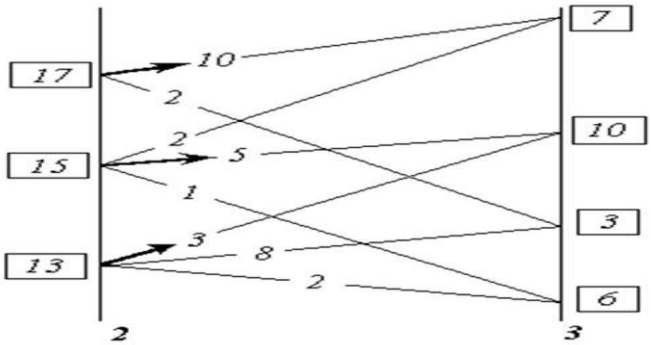


Рисунок 2 - Анализ переходов.

После этого множество «2», предшествовавшее завершающему процесс множеству «3», можно рассматривать в качестве завершающего, поскольку известны оценки каждого из его возможных состояний (максимальные полные выигрыши) и дальнейшая оптимизация последовательности шаговых управлений и выбор оптимальной траектории могут быть проведены только на ещё не рассмотренных множествах, предшествующих множеству «2» в оптимизируемом процессе (т.е. на множествах «0» и «1»).

Таким образом, процедура, иллюстрируемая рисунком 2, работоспособна на каждом алгоритмическом шаге метода при переходах из n—го в $(n - 1)$—е множество, начиная с

завершающего N—ного множества до начального состояния системы.

В результате последовательного попарного перебора множеств, при прохождении всего их набора, определяется оптимальная последовательность преемственных шаговых управлений, максимально возможный полный выигрыш и соответствующая им траектория. На рисунке 2 утолщённой линией показана оптимальная траектория для рассматривавшегося примера.

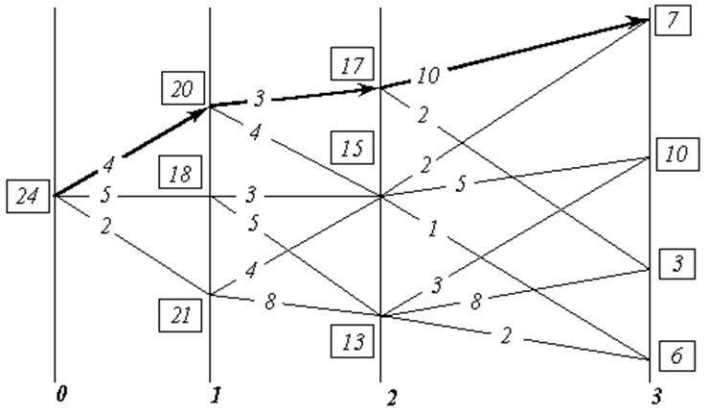


Рисунок 3. Оптимальная траектория.

В рассмотренном примере критерий оптимальности – сумма шаговых выигрышей. Но критерий оптимальности может быть построен и как произведение *обязательно неотрицательных* сомножителей.

Поскольку результат (сумма или произведение) не изменяется при изменении порядка операций со слагаемыми или сомножителями, то алгоритм работоспособен и при переборе множеств возможных состояний в порядке, обратном рассмотренном (от исходного к завершающему множеству возможных состояний).

Если множества возможных состояний упорядочены в хронологической последовательности, то это означает, что

расчетная схема может быть построена как из реального настоящего в прогнозируемое *определённое* будущее, так и из прогнозируемого *определённого* будущего в реальное настоящее.

Это обстоятельство говорит о двух неформальных соотношениях реальной жизни, лежащих вне алгоритма:

1. Метод динамического программирования формально алгоритмически нечувствителен к характеру причинно-следственных обусловленностей (в частности, он не различает причин и следствий). По этой причине каждая конкретная интерпретация метода в прикладных задачах должна строиться с неформальным учетом реальных обусловленностей следствий причинами.
2. Если прогностика в согласии с иерархически высшим объемлющим управлением, а частное вложенное в объемлющее управление осуществляется квалифицировано, в силу чего процесс частного управления протекает в ладу с иерархически высшим объемлющим управлением, то не существует управленчески значимой разницы между ними.

Для пользования методом динамического программирования и сопутствующими его освоению неформализованными в алгоритме *жизненными проявлениями матриц перехода*, необходимо соблюдение главного из условий:

В задачах оптимизации процессов управления метод динамического программирования «реального будущего: – по умолчанию» работоспособен только, если определён вектор целей управления, т.е. должно быть избрано завершающее процесс.

В реальности это завершающее определённое состояние должно быть заведомо устойчивым и приемлемым процессом, объемлющим и несущим оптимизируемый методом частный процесс.

Неспособность определить вектор целей управления (достижением которого должен завершиться оптимизируемый в методе процесс) и (или) неспособность выявить исходное состояние объекта управления не позволяют последовать этой рекомендации, что *объективно закрывает* возможности к использованию метода динамического программирования, поскольку начало и конец процесса должны быть определены в пространстве параметров, на которых построена математическая (или иная) модель метода, которая должна быть метрологически состоятельной, что является основой её соотнесения с реальностью. Причём определённость *завершения* оптимизируемого процесса имеет управленчески большее значение, чем ошибки и некоторые неопределённости в идентификации (выявлении) начального состояния объекта управления.

Концепция управления в объективной мере, обладает собственными характеристиками, которые совместно с субъективными характеристиками субъекта-управленца, порождают вероятностную предопределённость осуществления им концепции управления. Значение вероятностной предопределённости успешного завершения процесса – объективная иерархически высшая мера, оценка замкнутой системы «объект + управленец + концепция», в отличие от вероятности – объективной меры системы «объект + объективно существующая концепция управления».

Поэтому, чем ниже вероятность перевода объекта в желательное завершающее состояние, тем выше должна быть квалификация управленца, повышающая значение вероятностной предопределённости успешного завершения процесса управления.

Таким образом, для администратора признание им некой концепции управления может выражаться в его уходе с должности по собственной инициативе, проистекающей из осознания им своей

неспособности к осуществлению признанной им концепции управления; а неприятие концепции может выражаться, как заявление о её принятии и последующие искренние ревностные, *но неквалифицированные* усилия по её осуществлению. Они приведут к тому, что концепция будет дискредитирована, поскольку квалифицированные управленцы, способные к её осуществлению, не будут допущены до управления по личной ревности, жажде славы, зарплаты или ещё чего-то со стороны благонамеренного самонадеянного неквалифицированного недочеловека.

То есть, необходимо поддерживать достаточно высокое качество управления на каждом шаге всеми средствами, чтобы не оказаться к началу следующего шага в положении, из которого в соответствии с избранной концепцией управления перевод объекта в избранное завершающее состояние невозможен.

Метод динамического программирования эффективнее применять к решению задач с дискретными переменными, чем с непрерывными и позволяет найти все оптимальные решения, аддитивной функции многих переменных на любом множестве, заданными условиями-ограничениями. В случае, если таких ограничений не велико, задача динамического программирования может быть сведена к решению задачи линейного программирования.

К наиболее типовым задачам, разрешимым с помощью метода динамического программирования, в области оптимизации управления в здравоохранении можно отнести, например: оптимальная загрузка транспортного средства неделимыми предметами, о вкладе страховых средств в лечебно-оздоровительные учреждения, транспортная задача, поиск экстремума в приборах, анализирующих множество показателей состояния биообъекта.

Порядок выполнения работы:

Специфика выполнения данной лабораторной работы по изучению основ динамического программирования заключается в изучении в сети Интернет определенных тем и программных пакетов, реализующих метод динамического программирования.

1. Подготовить краткое сообщение по одной из следующих тем (определяется согласно варианту задания):

- Решение задачи оптимизации методами динамического программирования потоков посетителей лечебно-оздоровительных учреждений в регионе.
- Решение задачи оптимизации методами динамического программирования комплектации средствами лечения с различных баз в регионе.
- Решение задачи повышения качества диагностики в процессе итерационного вычисления критериев качества
- Самоорганизационные алгоритмы решения задач динамического программирования.
- Сравнительный анализ алгоритмов реализации методов линейного и динамического программирования.

2. Найти в Интернет варианты программного обеспечения метода динамического программирования (не менее 3) и выполнить сравнительный анализ.

3. Решить методом динамического программирования следующую задачу: имеется маршрут из 10 городов, соединенных дорогами, передвижениями по которым определяются определенными материальными затратами. Требуется найти путь, по которому наиболее оптимально доставить больного из города А в город Б с минимизацией материальных затрат, связанными со скоростью передвижения. Граф связности, определяющий передвижение из города А в город Б определяется следующим образом:

- из всего множества городов (10) случайным образом определяются начальный и конечный город;
- случайным образом строится граф связности между остальными (восьмью) городами i и j путем генерации чисел в диапазоне [-min(ii,jj),max(ii,jj)];

- в случае отрицательного числа – связь между городами отсутствует;
- в случае положительного числа – случайным образом для каждого соединения определяются длинна пути и скорость передвижения кареты «скорой помощи»;
- случайным образом определяются первые три и последние два города (первые – в которые попадаем из город А, последние - из которых попадаем в город Б);
- зафиксировать полученный путь графически и в виде матрицы связности, весами которых будут времена передвижения «скорой помощи» между городами (вершинами графа).

4. Оформить отчет

Контрольные вопросы:

1. Что называется динамическим программированием?
2. В чем состоит сущность динамического программирования?
3. Как формулируется и решается транспортная задача в динамическом программировании?
4. Каким образом ищется экстремальное значение целевой функции методами динамического программирования?
5. Какие ограничения в применении имеет метод динамического программирования?
6. При каких условиях функционирования системы управления БТС рекомендуется применять идеологию динамического программирования?

Библиография.

1. Грешилов А.А. Математические методы принятия решений: Учеб. пособие для вузов. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2016.-584 с.
2. Динамическое программирование. Презентация on-line / http://ppt-online.org/8147
3. Оптимизация прикладных задач. Вводный курс [Текст] : учебник / П. Н. Учаев [и др.] ; ред. П. Н. Учаев. - Старый Оскол : ТНТ, 2014. - 288

4. Построение математической модели и решение транспортной задачи [Электронный ресурс]: методические указания по выполнению лабораторной работы по дисциплине «Математическое моделирование статических и динамических объектов и процессов» для студентов направления подготовки 211000.62 / Юго-Западный государственный университет, Кафедра конструирования и технологии электронно-вычислительных средств; ЮЗГУ; сост. Л. А. Болычевцева. -Курск:ЮЗГУ,2014. -19с
5. Презентация на тему Динамическое программирование / http://presentaci.ru/prezentacii-po-programmirovaniu/7442-dinamicheskoe-programmirovanie.html
6. Сачков, В. Н. Введение в комбинаторские методы дискретной математики [Текст] / В. Н. Сачков. - М.: МЦНМО, 2014. - 424 с.
7. Устюжанин, В. А. Моделирование биотехнических систем [Текст]: учебное пособие / В. А. Устюжанин, И. В. Яковлева. - Старый Оскол: ТНТ, 2014. - 216 с.
8. Ченцов, А. Г. Экстремальные задачи маршрутизации и распределения заданий: вопросы теории [Текст] / А. Г. Ченцов. - М.; Ижевск: Регулярная и хаотическая динамика, 2018. - 240 с

5. ВЫДЕЛЕНИЕ РИТМИЧЕСКИХ СОСТАВЛЯЮЩИХ МЕДИКО-БИОЛОГИЧЕСКИХ СИГНАЛОВ

Цель работы: изучение методов и овладение навыками применения ЭВМ для выделения и анализа колебательных (ритмических) составляющих в медико-биологических сигналах, характеризующих автономные системы управления различных биообъектов и систем.

Краткие теоретические сведения.

Хорошо известно, что биологические системы обладают свойствами саморегуляции, то есть способностью перестраиваться в зависимости от внешних воздействий так, чтобы сохранился оптимальный уровень их функционирования.

Существуют различные способы регуляции жизнедеятельности клетки, которые можно условно отнести к генетическому, биохимическому и физиологическому уровням регуляции. В пределах каждого из них действуют механизмы, в основе которых лежит последовательность конкретных метаболических процессов. Понять динамические свойства этих регуляторных механизмов можно лишь на основе общесистемного подхода, рассматривающего поведение каждого из элементов сложной системы как результат его взаимодействия с остальными элементами.

Одним из наиболее развитых подходов для решения этой проблемы в современной биофизике является математическое моделирование. В соответствующих кинетических моделях отражена динамика изменения концентраций различных составных элементов биологической системы, которая определяется скоростями отдельных элементарных реакций.

Биологические системы – это объекты различной сложности, имеющие несколько уровней структурно-функциональной организации и представляющие собой совокупность взаимосвязанных и взаимодействующих элементов. Примерами биологических систем являются: клетка, ткани, органы, организмы,

популяции, виды, биоценозы, экосистемы разных рангов и биосфера.

Ритмичность – свойство, присущее как живой, так и неживой природе. Оно обусловлено различными космическими и планетарными причинами: вращением Земли вокруг Солнца и вокруг своей оси, фазами Луны и т.д. Ритмичность проявляется в периодических изменениях интенсивности физиологических функций и формообразовательных процессов через определенные равные промежутки времени. Хорошо известны суточные ритмы сна и бодрствования у человека, сезонные ритмы активности и спячки у некоторых млекопитающих и многие другие. Ритмичность направлена на согласование функций организма с периодически меняющимися условиями жизни.

Для биологических систем характерно периодическое изменение различных характеристик. Период этих колебаний может быть связан с периодическими изменениями условий жизни на Земле смена времен года, смена дня и ночи. Существуют и другие геофизические ритмы –солнечные, лунные, связанные с периодами атмосферных явлений. Геофизические и биологические ритмы сопоставлены на рисунке1. Но многие периодические процессы имеют частоту изменения, не связанную очевидным образом с внешними геокосмическими циклами. Это так называемые «биологические часы» различной природы, начиная от колебаний биомакромолекул, биохимических колебаний, вплоть до популяционных волн.

Внутриклеточные колебания задают эндогенные биологические ритмы, которые свойственны всем живым системам. Именно они определяют периодичность деления клеток, отмеряют время рождения и смерти живых организмов. Модели колебательных систем используются в ферментативном катализе, теории иммунитета, в теории трансмембранного ионного переноса, микробиологии и биотехнологии.

С некоторыми из типов периодических движений мы уже имели дело при рассмотрении особых точек типа центр и затухающих или нарастающих колебаний в случае устойчивого и неустойчивого фокуса. Однако «биологические часы» имеют

свойство, отличающее их от рассмотренных типов колебаний - неизменность во времени периода и амплитуды таких колебаний, означающую стационарность и устойчивость колебательного режима.

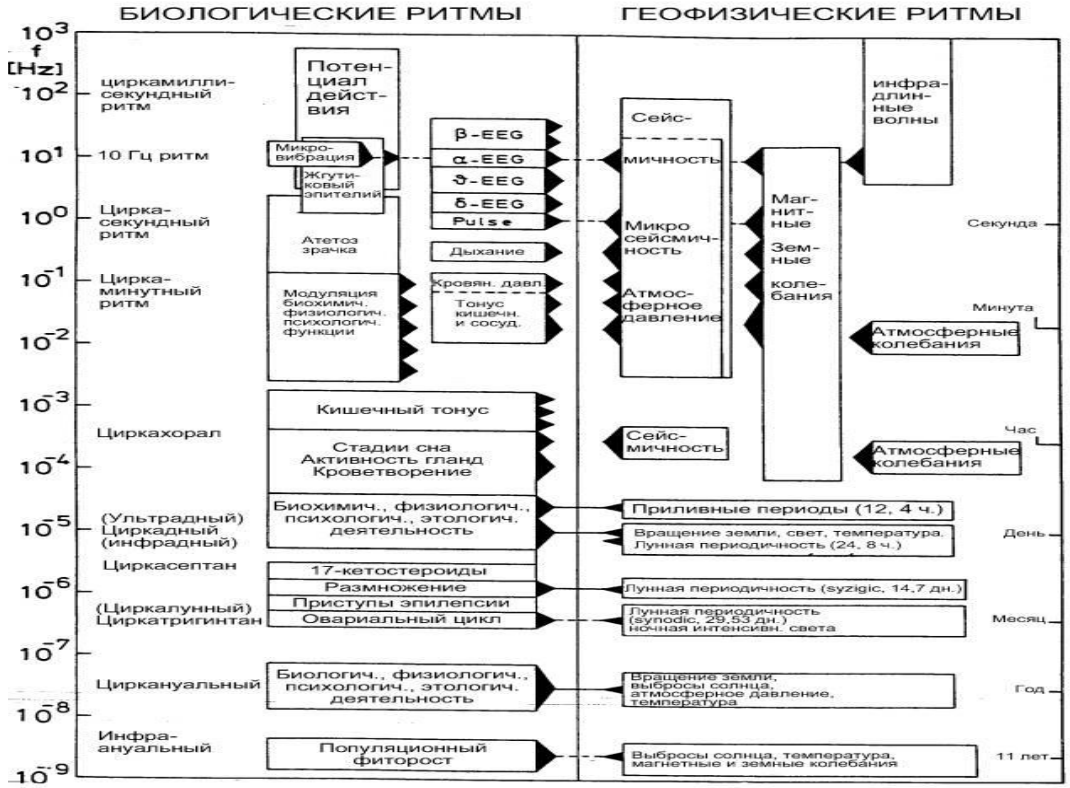


Рис. 1. Космофизические, геофизические и биологические ритмы. Справа – шкала периодов, слева – шкала частот

В данном случае периодическое изменение величин представляет собой один из типов стационарного поведения системы. Если колебания в системе имеют постоянные период и амплитуду, устанавливаются независимо от начальных условий и поддерживаются благодаря свойствам самой системы, а не вследствие воздействия периодической силы, система называется автоколебательной.

Незатухающие колебания в таких системах устойчивы, так как отклонения от стационарного колебательного режима затухают. К классу автоколебательных систем относятся колебания в гликолизе и других метаболических системах, периодические процессы

фотосинтеза, колебания концентрации кальция в клетке, колебания численности животных в популяциях и сообществах.

Предельный цикл. В фазовом пространстве такому типу поведения соответствует притягивающее множество (аттрактор), называемое предельным циклом. Предельный цикл есть изолированная замкнутая кривая на фазовой плоскости, к которой при $t \to \infty$ стремятся все интегральные кривые. Предельный цикл представляет стационарный режим с определенной амплитудой, не зависящий от начальных условий, а определяющийся только организацией системы. Существование предельного цикла на фазовой плоскости есть основной признак автоколебательной системы. Очевидно, что при автоколебательном процессе фаза колебаний может быть любой.

Остановимся на общих характеристиках автоколебательных систем. Рассмотрим систему уравнений общего вида:

$$\frac{dx}{dt} = P(x, y),$$
$$\frac{dy}{dt} = Q(x, y). \qquad (1)$$

Если T (T > 0) — наименьшее число, для которого при всяком t

$$x(t + T) = x(t), \qquad y(t + T) = y(t),$$

то изменение переменных x = x(t), y = y(t) называется периодическим изменением с периодом T.

Периодическому изменению соответствует замкнутая траектория на фазовой плоскости, и обратно: всякой замкнутой траектории соответствует бесконечное множество периодических изменений, отличающихся друг от друга выбором начала отсчета времени.

Если периодическому изменению на фазовой плоскости соответствует изолированная замкнутая кривая, к которой с внешней и внутренней стороны приближаются (при возрастании t) соседние траектории по спиралям, эта изолированная замкнутая траектория есть предельный цикл.

Простые примеры позволяют убедиться, что система общего вида (1) допускает в качестве траекторий предельные циклы.

Например, для системы

$$\frac{dx}{dt} = y + x[1-(x^2+y^2)],$$
$$\frac{dy}{dt} = -x + y[1-(x^2+y^2)], \qquad (2)$$

траектория $x^2 + y^2 = 1$ является предельным циклом. Его параметрические уравнения будут: $x = \cos(t - t_1)$, $y = \sin(t - t_1)$, а уравнения всех других фазовых траекторий запишутся в виде: $x = \frac{\cos(t - t_0)}{\sqrt{1 + Ce^{-2(t-t_0)}}}$, $y = \frac{\sin(t - t_0)}{\sqrt{1 + Ce^{-2(t-t_0)}}}$.

Значениям постоянной интегрирования C > 0 соответствуют фазовые траектории, накручивающиеся на предельный цикл изнутри (при $t \to \infty$), а значениям –1 < C < 0 траектории, накручивающиеся снаружи.

Предельный цикл называется устойчивым, если существует такая область (окрестность) на фазовой плоскости, содержащая этот предельный цикл, , что все фазовые траектории, начинающиеся в окрестности асимптотически при $t \to \infty$ приближаются к предельному циклу. Если же, наоборот, в любой сколь угодно малой окрестности предельного цикла существует по крайней мере одна фазовая траектория, не приближающаяся к предельному циклу при $t \to \infty$, то такой предельный цикл называется неустойчивым. Такие циклы разделяют области влияния (бассейны) разных притягивающих множеств.

Устойчивость предельного цикла (и устойчивость в смысле Ляпунова соответствующих периодических движений) определяется знаком характеристического показателя. Предельный цикл устойчив, если h < 0 и неустойчив, если h > 0. Если же h = 0, уравнения первого приближения не решают вопроса об устойчивости периодического движения.

Для нахождения предельных циклов не существует таких простых аналитических методов, как для нахождения стационарных точек и исследования их устойчивости. Однако, исследование фазовой плоскости системы позволяет ответить на вопрос, есть в данной системе предельный цикл, или нет.

Сформулируем несколько теорем, определяющих наличие предельного цикла по топологическому строению фазовой

плоскости. Они могут быть полезны как при аналитическом, так и при компьютерном анализе системы.

Теорема 1. Пусть на фазовой плоскости существует область, из которой фазовые траектории не выходят, и в которой нет положений равновесия (особых точек). Тогда в этой области обязательно существует предельный цикл, причем все остальные траектории обязательно наматываются на него.

Это означает, что фазовые траектории либо входят, пересекая границу, внутрь области, либо сама граница является траекторией. Легко видеть, что такая область не может быть односвязной. Поскольку траектория наматывается на предельный цикл изнутри, это означает, что внутри этого предельного цикла на фазовой плоскости существует либо неустойчивая особая точка, либо неустойчивый предельный цикл, очевидно, не принадлежащие рассматриваемой области G.

Таким образом, если найти на фазовой плоскости такую двусвязную область, что направления фазовых траекторий на всей границе обращены внутрь этой области, то можно утверждать, что внутри этой области имеется предельный цикл.

Теорема 2. Если существует на фазовой плоскости некоторая замкнутая область, такая, что все фазовые траектории, пересекающие границу этой области, входят в нее, и внутри этой области находится неустойчивая особая точка, то в этой области обязательно имеется хотя бы один предельный цикл.

Рассмотрим некоторые критерии отсутствия замкнутых фазовых траекторий (в том числе предельных циклов).

1. Если в системе не существует особых точек, то в ней не может быть и замкнутых фазовых траекторий.
2. Если в системе существует только одна особая точка, отличная от узла, фокуса и центра (например, седло), то такая система не допускает замкнутых фазовых траекторий.
3. Если в системе имеются только простые особые точки, причем через все точки типа узел и фокус проходят интегральные кривые, уходящие на бесконечность, то в такой системе нет замкнутых фазовых траекторий.

В случае, если критерии 1–3 выполнены, можно с уверенностью утверждать, что в системе нет предельных циклов. Однако невыполнение этих критериев еще не позволяет сделать вывод о наличии в системе предельных циклов и, следовательно, автоколебаний.

Неустойчивый предельный цикл также может содержаться в фазовом портрете грубых систем. Однако такой предельный цикл не соответствует реальному периодическому процессу, он играет лишь роль «водораздела», по обе стороны которого траектории имеют различное поведение. Например, представлять собой сепаратрису, отделяющую область тяготения траекторий к устойчивой особой точке, с одной стороны, и к устойчивому предельному циклу, с другой.

Рождение предельного цикла. Бифуркация Андронова-Хопфа. Существование предельных циклов возможно лишь в системе, правые части которой представлены нелинейными функциями. Нулевым значениям действительной части характеристических чисел (ляпуновских показателей) соответствует особая точка типа центр. В нелинейной системе, где возникает неустойчивый фокус, при этом возможно рождение предельного цикла. Такой переход легко проследить в «модельной» системе: $\frac{d\varphi}{dt} = 2\pi.\ \frac{dr}{dt} = r(c - r^2)]$,

Схематически возникновение предельного цикла в системе изображено на фазопараметрической диаграмме на рисунке 2.

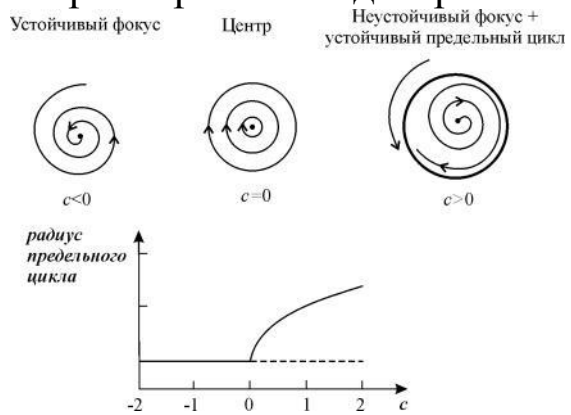


Рис. 2. Закритическая (суперкритическая) бифуркация Андронова-Хопфа.

Мягкое возбуждение. При c>0 возникают автоколебания, амплитуда которых растет с увеличением c. Бифуркация впервые была исследована А.А. Андроновым для случая N = 2 и обобщена Е. Хопфом на системы с произвольной размерностью. (Андронов А.А., Витт А.А., Хайкин С.Э. Теория колебаний. М., Наука, 1981; Hopf E., 1942). Существуют два типа бифуркации Андронова–Хопфа. Только что мы рассмотрели суперкритическую бифуркацию (мягкое возбуждение автоколебаний). Возможна также субкритическая бифуркация (жесткое возбуждение автоколебаний). В этом случае при бифуркационном значении параметра устойчивый фокус теряет устойчивость из-за «влипания» в него неустойчивого предельного цикла. Фокус становится неустойчивым, а аттрактором при этом может стать предельный цикл большой амплитуды.

«Модельной» системой, описывающей рождение предельного цикла при жестком возбуждении, является система:

$$\frac{dr}{dt} = r\,(c + r^2 - r^4)],$$
$$\frac{d\varphi}{dt} = 2\pi. \qquad (4)$$

Приравняв правую часть первого уравнения нулю, получим стационарные значения r: $r = 0,\ \ r^2 = 1 \pm (1+c)^{1/2}$. Ветвь r = 0 устойчива при c < 0 и неустойчива при c > 0. При c > –1 стационарное решение $r = [1 \pm (1+c)^{1/2}]^{1/2}$ □ устойчивый предельный цикл.

При –1 < c < 0 стационарное решение $r = [1 \pm (1+c)^{1/2}]^{1/2}$ образует неустойчивый предельный цикл.

Рассмотрим, что произойдет, если двигаться по параметру c, начиная с отрицательных значений. Первоначально имеется единственное устойчивое стационарное состояние r = 0, колебаний нет. При c > –1 существует также устойчивый предельный цикл, но система не покидает своего устойчивого стационарного состояния. Однако после того как *c* становится положительным, стационарное состояние становится неустойчивым, и происходит резкий скачок к устойчивому предельному циклу. В системе начинаются колебания

сразу большой амплитуды. Если двигаться от положительных значений с к отрицательным, колебания большой амплитуды сохраняются до тех пор, пока с не станет меньше –1, а затем внезапно исчезнут. Таким образом, при $-1 < c < 0$ могут существовать два различных типа поведения. Какой из них реализуется, зависит от предыстории системы.

Такой феномен называется эффектом г*истерезиса*.

При увеличении параметра *c* и его переходе через ноль скачком возникают устойчивые автоколебания конечной амплитуды и частоты. Для промежуточных значений параметра с существуют два типа устойчивого поведения (два аттрактора) - устойчивое стационарное состояние и устойчивый предельный цикл.

Винфри (Winfree A.T.) назвал области, в которых возможны два режима: устойчивая точка покоя и предельный цикл, «черной дырой». В этой области параметров можно так приложить возмущение к колебательной системе, что она попадет в область притяжения точки покоя, что приведет к прекращению колебаний. В частности, это показано для уравнений Ходжкина–Хаксли, моделирующих проведение нервного импульса (см. ниже).

Брюсселятор. Простейшим классическим примером существования автоколебаний в системе химических реакций является тримолекулярная модель «Брюсселятор», предложенная в Брюсселе Пригожиным и Лефевром (1965). Основной целью при изучении этой модели было установление качественных типов поведения, совместимых с фундаментальными законами химической и биологической кинетики.

В этом смысле блюсселятор играет роль базовой модели, такую же как гармонический осциллятор в физике, или модели Вольтерра в динамике популяций. Во 2-й части лекций мы остановимся на пространственно-временных свойствах распределенной системы, локальным элементом которой является брюсселятор. Здесь мы рассмотрим свойства брюсселятора как автоколебательной системы.

Брюсселятор содержит простейшую реализацию кубической нелинейности посредством химической реакции.

Хотя тримолекулярная стадия в химической кинетике не столь распространена, как бимолекулярные процессы, выражения для скорости ряда биохимических реакций в определенных случаях можно свести к кубическому виду. Брюсселятор представляет собой следующую схему гипотетических химических реакций:

$$A \underset{k_{-1}}{\overset{k_1}{\rightleftarrows}} X$$

$$2X + Y \underset{k_{-2}}{\overset{k_2}{\rightleftarrows}} 3X$$

$$B + X \underset{k_{-3}}{\overset{k_3}{\rightleftarrows}} Y + C$$

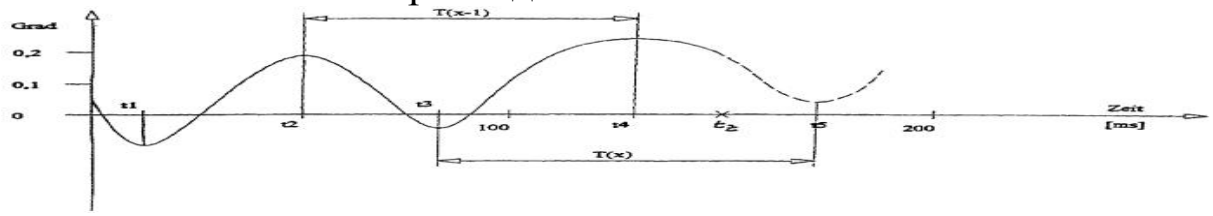
$$X \underset{k_{-4}}{\overset{k_4}{\rightleftarrows}} R$$

Здесь A, B — исходные вещества, C, R — продукты, X, Y — промежуточные вещества. Пусть конечные продукты C и R немедленно удаляются из реакционного пространства. Значения остальных констант положим равными единице. Тогда схема реакций (в случае точечной системы) описывается системой уравнений: $\frac{dx}{dt} = A + X^2 Y - (B+1)X$, $\frac{dy}{dt} = BX - X^2 Y.$

Модель имеет одну особую точку с координатами: $\bar{X} = A, \quad \bar{Y} = \frac{B}{A}.$

.

Существует множество *способов анализа колебаний биообъектов*. Рассмотрим один из нах.



Изобретение относится к способам исследования управления в биологических системах, применяемым в физиологии, медицине, психологии и биокибернетике. Способ анализа колебаний, произведенных в биологических системах, при котором изменение условий для вероятности появления случайного события можно

оценить по сравнению с предшествующим циклом благодаря вступлению энергетической составной части в текущий цикл колебания. Для этого анализируемый сигнал необходимо разбивать на последовательности циклов так, что каждый последующий цикл имеет общий временной интервал с предшествующим циклом. Изобретение относится к способам исследования управления в биологических системах, применяемым в физиологии, медицине, психологии и биокибернетике.

В физической биологии биологические системы характеризуются иногда как ансамбли взаимосвязанных и взаимно нагруженных осцилляторов. В соответствии с физической теорией гомеокинезиса А.С. Ибераля стабильная организация системы есть следствие взаимодействия осциллирующих процессов на всех уровнях системы (A.S Iberall: Toward a general science of viable systems, McGraw-Hill, New York 1972).

Некоторые сигналы, которые исходят от мозга, мышц или других частей тела и регистрируются электрическими, магнитными или механическими датчиками, являются следствиями такого рода осциллирующих активностей. Анализ колебаний, порождаемых осциллирующими активностями, позволяет получить сведения об особенностях протекания физиологических, психических и др. процессов, о взаимодействии этих процессов между собой. На основании полученных результатов оценивается и коррегируется текущее состояние системы.

В качестве примера анализа колебаний для таких целей могут привлекаться исследования тремора. Тремор - это непроизвольное ритмичное осциллирующее движение части тела, которое интерпретируется как ориентировочно синусоидальное движение, причем частота тремора у нормальных людей лежит в интервале 8-12 Гц. Тремор также используется как симптом, характеризующий нейромускульную систему.

Инициации дискретных произвольных движений обнаруживают тенденцию стартовать в области фазы физиологического тремора, которая обладает пиком скорости в направлении этого произвольного движения. Такого рода произвольные движения могли бы облегчаться посредством

тремора (D.Goodman, J.A.Kelso: Exploring the functional significance of physiological tremor: A biospectroscopic approach, Exp. Brain Res. 49, S 419-431, 1983).

У пациентов, страдающих болезнью Паркинсона, предпочтительная фаза инициации дискретных произвольных двигательных ответов находится в области тремора, которая обладает пиком скорости в направлении, противоположном этому двигательному ответу. Если произвольная инициация двигательного ответа выполняется как реакция на сигнал, то латентное время реакции зависит от фазы инициации в цикле тремора (G. Staude u.a.: Tremor as a factor in prolonged reaction times of Parkinsonian patients, Movement Disorders Vol.10, Nr.2, S 153-162, 1995).

Для получения ранее указанных данных о взаимодействии тремора с двигательным выходом системы исследователи разделяли треморограммы на периоды между соседними максимумами. Фаза инициации в цикле тремора, который нарушен двигательным ответом, вычислялась на основании ее длины из предшествующего цикла тремора. При этом не учитывалось, что вероятность появления сигнала во время длинных циклов тремора выше, чем во время коротких циклов тремора, если сигналы появляются по случайному закону. Экспериментально подтверждено (G.Staude; W.Wolf: Voluntary motor reactions: does stimulus appearance prolong the actual tremor period, Journal of Electromyography and Kinesiology 9, S.277-281, 1999) увеличение средней длительности цикла тремора, определенного между локальными максимумами треморограмм, по сравнению со средней длительностью каждого из двух таких же предшествующих циклов тремора. Остается вопрос, рассматривать ли увеличенное статистическое среднее значение длительности цикла как ошибку способа определения средней оценки или рассматривать как признак механизма управления, с помощью которого познаваема стабильная организация жизнеспособных систем.

Доказано, что некоторые частоты, участвующие в формировании структуры тремора происходят из центральной нервной системы. На основании этого факта можно предположить,

что способы изменения соотношений длительностей двух соседних энергетических частей между переходами первой производной через ноль, которые могут представлять различные циклы треморов, содержат информацию о признаках управления так же, как способы изменения соотношений длительности циклов.

При ответе на поставленный выше вопрос об увеличенном статистическом среднем значении длительности цикла следует учитывать, что момент появляющегося сигнала принадлежит как периоду между соседними максимумами, так и периоду между соседними минимумами. Длительности этих периодов можно интерпретировать как результат взаимных нагружений взаимосвязанных осцилляторов. Тогда разность длин названных периодов характеризует изменение вероятности появления случайного события всегда во второй энергетической части последующего периода по сравнению с вероятностью его появления во второй энергетической части предшествующего периода. Следовательно, признаки управления изменениями вероятности для «захвата» ожидаемого события структурами колебания следует искать в разности длительности не общих частей, которые меньше чем период. Похожим образом время инверсии τ для „time invariance" теста выбирают в пределах временного интервала между локальным максимумом и предшествующим локальным минимумом, чтобы получить высшее значение асимметрии и по величине асимметрии различать принадлежность стратегии процессов формирования волн к определенным треморам (G.Deuschl u.a.: Tremor classification and tremor time series analysis, Chaos, Nr.5(1), S.48-52, 1995).

Для того чтобы наблюдаемые изменения вероятностей отнести к управлению, нужно исследовать особенности поведения длительности другим образом построенных циклов или составных частей циклов, измеренных между локальным максимумом и предшествующим локальным минимумом и/или в обратной последовательности. Регистрация ускорений тремора кисти в одном и том же направлении проводится посредством ON/OF датчика методом инструментального измерения нейропсихофизического состояния пациента по EP 1095617, что соответствует определению

только одной составной части цикла тремора. Однако эти данные используются далее для оценки интенсивности тремора за исследованный промежуток времени.

В связи с этим возникает задача разработать способ анализа колебаний в биологических системах, при котором изменение условий для вероятности появления случайного события можно оценить по сравнению с предшествующим циклом благодаря вступлению очередной энергетической составной части в текущий цикл колебания. Получение такой информации возможно в случае, если интерпретировать каждый временной интервал между локальным максимумом и предшествующим соседним минимумом (или в обратной последовательности) как составляющие единое целое две составные части двух циклов.

Эта задача решается посредством реализации отличительных признаков изобретения. Новый способ анализа колебаний, произведенных биологическими системами, устраняет названные выше недостатки предшествующего уровня техники.

Преимущество способа, соответствующего изобретению, заключается в том, что посредством различных анализов изменения длительности следующих друг за другом циклов или определенных временных интервалов возможно обнаруживать стратегии управления и их изменение от измененных условий.

Способ разделения анализируемого сигнала на последовательность циклов, из которых получают длительность параметров для дальнейших анализов, описывается более подробно на основании примера исполнения. Для этого отдельный чертеж показывает образец треморограммы (чертеж).

В треморограмме, показанной на чертеже, анализируемая кривая перед появлением случайного сигнала в момент времени tz изображена сплошной линией, а после появления - штриховой линией.

Предложенный способ анализа колебаний в биологических системах найдет широкое применение в различных областях науки:

- исследование биологических механизмов переработки информации,
- определение общих и индивидуальных способов адаптации,

- индивидуальные тестирования для понимания действительных способностей, например, у спортсменов,
- определение состояния и управления тренировочной нагрузкой у высококвалифицированных спортсменов,
- оценки эффективности настройки на выполняемое действие и корректировка этой настройки,
- выбор команды и/или частей команды (группы людей),
- исследование реакции у животных на изменение природных условий, измерение и понимание этих реакций.

1. Способ анализа колебаний, произведенных в биологических системах, при котором колебания, преобразованные в усиленный сигнал для дальнейших измерений, запоминаются вместе с раздражителями за время регистрированного колебания, в котором при воспроизведении для анализа сигналы фильтруются, устанавливаются участки для анализа, в участках определяются локальные максимумы равно, как локальные минимумы, и нумеруются по порядку, измеряются временные интервалы между каждым локальным минимумом и ближайшим соседним максимумом и между каждым локальным максимумом и ближайшим соседним минимумом, отличающийся тем, что последовательности циклов соединяются так, что каждый цикл состоит из двух вышеназванных соседних временных интервалов, и что анализируемый сигнал так разделяется на последовательность циклов, что каждый следующий цикл имеет общий временной интервал с предшествующим соседним циклом.
2. Способ по п.1, отличающийся тем, что разности соседних временных интервалов соседних циклов используются в качестве анализируемых параметров.
3. Способ по п.2, отличающийся тем, что отношения соседних временных интервалов соседних циклов используются в качестве анализируемых параметров.

Порядок выполнения работы:

1. Выделить из исходного массива данных (см. Приложение) два вектора пульсограмм (до и после нагрузки) длинной 64 измерений,

начиная с номера, определенного датчиком случайных чисел в интервале от 1 до вашего номера в студенческом журнале.

2. Построить карелограммы и определить функциональные зависимости между значениями ее координат. Оценить адекватность полученных моделей. Считая построенную каррелограмму фазовым портретом оценить устойчивость функционирования до и после нагрузки (для большей наглядности выполните построение на одной плоскости)

3. Выполните прямые и обратные преобразования Фурье первого и второго порядка – постройте графики их характеризующие (модуль, сдвиг фаз, мнимою и действительные части, годограф) до и после нагрузки, сделайте выводы.

4. Выполните численное дифференцирование пульсограмм, предположите частоты работы пейсмейкеров (2-3 частоты), идентифицируйте параметры колебательной модели анализируемых биосигналов модели по некратным частотам и оцените их адекватность (по средне-квадратичной ошибки отклонений).

5. Постройте по колебательным моделям, полученным в п.4 и п.5 (для спектра Фурье выберите 3-4 частоты с максимальной амплитудой) траектории движения биообъекта в пространстве запаздываний по времени dt=1, 2, 5, 10 (по оси абсцисс отложите значения модели в моменты времени t, по оси ординат – (t+dt). Проанализируйте устойчивость.

6. Возьмите лист бумаги (А4), вытяните руки и положите на них лист. Сделайте видиозапись тремора с открытыми и закрытыми глазами. Зафиксируйте с помощью секундомера частоты колебаний. По результатам нескольких экспериментов в группе выполните структурно параметрическую идентификацию модели функциональной зависимости между тремором с закрытыми и открытыми глазами. Сделайте выводы.

Контрольные вопросы:

1. Почему в биологических системах наблюдаются колебательные процессы?
2. Что такое автоколебания? Как оно связано с эффектом резонанса?
3. Что характеризует аттрактор?

4. Что характеризует фазовый портрет колебательной системы?
5. Когда возникают колебания в системах гистерезисного типа?
6. Для чего применяют прямые и обратные преобразования Фурье?
7. Каким образом, численное дифференцирование позволяет оценить присутствие колебательных процессов в биосигнале?

Библиография

1. Исследование спектров сигналов [Электронный ресурс] : методические указания по выполнению лабораторной работы № 1 по курсу «Общая теория связи» / Юго-Западный государственный университет, Кафедра телекоммуникаций; ЮЗГУ; сост.:С. Г. Лукьянюк, И. Г.Бабанин,С. С.Хотынюк. -Курск: ЮЗГУ, 2013. - 12 с.
2. Колебательные процессы в биологии. Презентация / http://mathbio.ru/lectures/2013l08p1/
Стрелков, С. П. Введение в теорию колебаний [Текст] : учебник / С. П. Стрелков. - 3-е изд., испр. - СПб. : Лань, 2005. - 440 с.
3. Плюснина Т.Ю., Фурсова П. В., Тёрлова Л. Д., Ризниченко Г. Ю. Математические модели в биологии . Учебное пособие. М.: -Ижевск: НИЦ: «Регулярная и хаотическая динамика», 2014. 136 с.
4. Ризниченко Г. Ю. Лекции по математическим моделям в биологии Издательство РХД, 2011 . -560с
5. Цифровая обработка сигналов и изображений в радиофизических приложениях. / Под. Ред. В.Ф. Кравченко. –М.: ФИЗМАТЛИТ, 2007.- 544 с.

Приложение к лабораторной работе:

Интервалограмма (пульсограмма)

до нагрузки

0,90	0,92	0,96	0,96	0,96	0,92	0,90	0,84	0,88	0,88	0,86	0,90	0,82
0,76	0,74	0,72	0,73	0,82	0,94	1,00	0,98	0,96	0,94	0,96	0,98	0,98
0,96	0,96	1,02	0,98	0,90	0,96	1,00	0,90	0,90	0,98	0,90	1,04	1,02
0,96	0,94	1,00	1,00	0,96	1,04	1,02	1,02	1,04	1,02	1,02	1,02	0,96
0,86	0,80	0,82	0,94	0,90	0,98	0,94	0,94	0,96	0,96	0,98	1,02	1,00
0,98	0,92	1,08	0,98	0,92	1,02	1,02	0,96	1,02	1,10	0,98	1,06	1,04

после нагрузки 1

0,64	0,72	0,74	0,76	0,78	0,76	0,80	0,76	0,74	0,72	0,72	0,72	0,76
0,78	0,76	0,80	0,74	0,78	0,76	0,74	0,74	0,84	0,80	0,80	0,82	0,74
0,76	0,80	0,78	0,80	0,78	0,74	0,74	0,72	0,70	0,64	0,62	0,64	0,66
0,74	0,76	0,72	0,78	0,76	0,72	0,74	0,70	0,70	0,74	0,76	0,74	0,80
0,84	0,76	0,78	0,72	0,72	0,70	0,72	0,72	0,76	0,80	0,74	0,80	0,80
0,74	0,76	0,74	0,74	0,74	0,76	0,76	0,80	0,82	0,78	0,80	0,80	0,78

6. ВЫДЕЛЕНИЕ ИНФОРМАТИВНЫХ ХАРАКТЕРИСТИК ИЗ МЕДИКО-БИОЛОГИЧЕСКИХ СИГНАЛОВ

Цель работы: овладение навыками формирования множества информативных признаков, характеризующих состояния биообъектами, на основе статистических характеристик.

Краткие теоретические сведения

В процессе обработки результатов исследования поведения биообъекта с целью построения решающих диагностических правил или иных исследований возникает проблема разрешения противоречия между многомерностью исходного признакового пространства и стремлением упростить модель для выделения наиболее существенных связей. Накопленный опыт информационного анализа свидетельствует о том, что любые статистические операции становятся более эффективными при уменьшении числа используемых признаков. Снижение размерности исходного пространства состояний осуществляется двумя основными путями:

- отбор признаков удовлетворяющих определенному критерию или целевой функции;
- заменой групп признаков на определенный функционал от данной группы.

Наиболее информативный признак обладает следующими свойствами:

1) он наиболее представителен в классе (свойство удовлетворяющей исследователя диагностической специфичности и чувствительности);
2) он наименее линейно зависим с другими (свойство ортогональности признакового пространства);
3) он достаточно вариативен как внутри класса, так и вне его (свойство репрезентативности).

Большинство алгоритмов формирования множества информативных признаков путем применения алгоритмов селекции работают в двух режимах: с заданием количества оставленных признаков (режим А) и с автоматическим отбором без априорно

заданного количества (режим Б). Все алгоритмы селекции могут работать как с первичным признаковым пространством, так и с уже обработанным (т.е. после одного и более режимов селекции).

К наиболее простым статистическим методам селекции признаков относятся:

- По относительному дисперсионному разбросу (вариации);
- По парной корреляции;
- По решаемым спорам;
- По качественной вариации;
- По информационному показателю силы влияния (ИПВ);
- По критериям различий выборок в различных классах (анализ доверительных интервалов, коэффициент Стьюдента и т.п.);
- По информационным критериям (Кульбака, анализ энтропийных характеристик и т.п.);
- По комплексным критериям, представляющие собой аддитивную, мультипликативную, дробно-рациональную, логическую или иные формы интеграции критериев различной семантики.

Перед проведением селекции по указанным критериям необходимо провести процедуры удаления артефактов и восстановления пропущенных значений.

Например, для восстановления пропущенных значений в признаках в случае их численного представления рекомендуется использовать метод линейной экстраполяции по двум ближайшим соседям, выбор которых осуществляется с учетом принципа несмещенности ошибки.

Математический аппарат механизмов селекции заключается в следующем:

- *По относительному дисперсионнуму разбросу (вариации).*

Каждый признак характеризуется величиной $V = 100 * D / E$, где D, E, V - соответственно, оценки среднеквадратичного отклонения, математического ожидания и вариации признака.

В режиме «А» пользователь задает N_{sel} - количество признаков, которое необходимо оставить после процедуры отбора.

После определения V_i , i=1,koolpr , строится гистограмма, по которой, начиная с верхней границей, подсчитывается количество признаков. При достижении границы N_{sel} осуществляется при

необходимости процедура удаления признаков, не отвечающих требованию селекции, с использованием датчика случайных чисел равномерного закона распределения. В режиме «Б» определяется пороговое значение показателя V; оставляются те признаки, у которых найденное Vi превышает указанное значение.

-По парной корреляции.

В связи с большой размерностью исходного признакового пространства в данном случае предлагается отойти от принципа построения полной матрицы корреляционной связи, осуществление которого приведет к возникновению проблемы «проклятия многомерности». Предлагается следующий механизм селекции.

Случайным образом, используя ДСЧ с равномерным законом распределения, строятся вектора:

R[k,i] , k=1,2,3 i=1,koolpr,

где R[k,i] - ранжированное с точностью до 0.1 абсолютное значение коэффициента парной корреляционной связи между признаками k и i.

Селекция по желанию пользователя осуществляется в 3 этапа:

-выбираются те факторы, у которых R[1,i] & R[2,i] & R[3,i] <=Rпор.;

-выбираются те факторы, у которых по i два из трех <=Rпор.;

-выбираются те факторы, у которых по i хотя бы один из трех <=Rпор.

В режиме А (задано N_{sel}) последовательно повторяются три этапа с определенным пороговым значением коэффициента парной корреляции (рекомендуется Rпор=0.14), а затем, с использованием ДСЧ, селектируются «лишние» признаки.

-По решаемым спорам.

Концепция этого метода заключается в том, что отбираются те признаки, которые имеют наименьшее количество «споров» - перекрытий доверительных интервалов по классам состояний биообъекта. Алгоритм в данном случае заключается в следующем.

Для каждого признака определяется доверительный интервал с определенным уровнем статистической значимости: $M_k \mp T_k$, где M_k- оценка модального значения признака, T_k – «трубка допуска» в k-ом классе.

Затем определяются величины G_{min} и G_{max} - минимальная и максимальная границы доверительных интервалов в различных анализируемых классах. Весь диапазон [G_{min},G_{max}] разбивается на равностоящие диапазоны, на которые «накладываются» интервалы признака из разных классов: если в диапазоне нет пересечения, то он приобретает значение «номера класса», в противном случае - номера «очередного класса». Т.о., определяется количество «новых» классов и диапазоны «перекрытия».

Если задано количество селектируемых признаков (режим «А»),то для каждого признака определяется значение критерия: $K_r = \frac{N_{kn} \cdot \sum_{i=1}^{N_k} \frac{\max(l,ll)}{\min(l,ll)}}{N_k}$, где N_{kn} - количество "новых" классов, N_k - первоначальное количество классов, l,ll -соответственно «длины» двух соседних классов в зонах перекрытия.

Суммирование ведется, если min(l,ll)<>0 и max(l,ll) не больше N_k.

Затем отбирается N_{sel} признаков с минимумом K_r.

В режиме «Б» пользователь задает максимально допустимую величину перекрытий и отбираются те признаки, у которых полученное количество перекрытий не превышает порога.

-По комплексному критерию (частный случай).

Данный метод аналогичен во многом описанному ранее механизму селекции по корреляции, но используется одновременно два критерия:

$V_{порог}$ и $R_{порог}$ (количественная вариация и корреляция).

Для каждого признака определяется значение критерия: $K_i = V_i * \left(\frac{1}{R[1,i]} + \frac{1}{R[2,i]} + \frac{1}{R[3,i]} \right)$, где $R[1,i]$, $R[2,i]$, $R[3,i]$ - три наименее коррелированных с признаком *i* значений коэффициентов корреляции.

В режиме «Б» задаются значения $V_{порог}$ и $R_{порог}$; оставляются те признаки, у которых $K_i \geq 3 \cdot V_{порог} / R_{порог}$.

В режиме «А» оставляется заданное количество признаков с максимальным значением K_i согласно построенной гистограмме этих значений и применения процедуры случайного выбора.

-По качественной вариации.

Данный вид отбора подобен селекции по количественной вариации, т.е. в качестве информативных отбираются обладающие максимальным значением коэффициента качественной вариации, но не более N_{sel} признаков. Коэффициент качественной вариации оценивается по формуле: $Q_i = \frac{l \cdot (N^2 - B)}{(l-1) \cdot N^2}$, где N - количество объектов, l - количество классов-диапазонов, B - сумма квадратов числа попаданий значений признаков в классы-диапазоны.

Примечание: Применять данный критерий следует осторожно, т.к. он несколько отражает свойства признака в смысле деления на классы, сколько говорит о равномерности или неравномерности распределения значений признака по классам-диапазонам. Т.е. в случае большой зашумленности наилучшими будут признаки с наиболее приближенным распределением частот своих значений к равномерному закону распределения.

-По информационному показателю силы влияния (ИПВ).

Данный критерий отсева основывается на свойствах информационной энтропии и является, пожалуй, наиболее эффективным, хотя и наименее быстрым из представленных ранее. Для каждого признака i строится гистограмма его значений, по которой оценивается общая энтропия $Э_i$. Поступая аналогичным образом в каждом классе, получаем внутриклассовые энтропии $Э_k$. Тогда негэнтропия равна: $NE_i = 1 - Э_i / \max(Э_k)$.

Отсюда имеем критерий: $ИПВ_i = NE_i / Э_i$.

Селектируются N_{sel} признаков с максимальными значениями ИПВ.

Порядок выполнения лабораторной работы

1. Сформировать множество базовое множество признаков согласно варианту (по таблицам приложения). Сформировать обучающую и экзаменационную последовательности равных объемов методом рандомизации (Монте-Карло).
2. Произвести выбор информативных признаков на обучающей выборке различными способами и сравнить результаты.

3. Произвести выбор информативных признаков на экзаменационной выборке различными способами и сравнить результаты.
4. Сравнить результаты п.2 и п.3
5. Оформить отчет.

Контрольные вопросы

1. Какие диагностические критерии используются при селекции наиболее информативных признаков?
2. Какие статистические критерии используются при селекции наиболее информативных признаков?
3. Какие интеллектуаризированные программные средства используются для формирования информативного признакового пространства?
4. Каким образом влияет изменение мерности признакового пространства на качество диагностики?
5. Как использовать доверительные интервалы для оценки информативности признаков?
6. Как использовать доверительные интервалы для оценки риска их использования в качестве инструмента формирования множества репрезентативных признаков?

Библиография

1. Демидова Л.А., Кираковский В.В., Пылькин А.Н. Принятие решений в условиях неопределенности. – М.: Горячая линия – Телеком, 2015. – 283 с.
2. Загоруйко Н.Г., Кутненко О. А. Алгоритм GRAD для выбора информативного подпространства признаков Институт Математики СО РАН, (точка доступа http://pandia.ru/text/78/248/79351.php)
3. Колесникова, С.Н. Методы анализа информативности разнотипных признаков.// Вестник Томского государственного университета. Управление, вычислительная техника и информатика. – № 1 (6). – 2009, – стр. 48-54.
4. Методы анализа информативности разнотипных признаков./URL: http://cyberleninka.ru/article/n/metody-analiza-informativnosti-raznotipnyh-priznakov
7. Саттон Р.С., Барто Э.Г. Обучение с подкреплением . – М: БИНОМ. Лаборатория знаний, 2011. – 369 с.

Приложение

Таблица 1 Загрязнения города 1 (фрагмент мониторинга, у.е.)

годы	Пыль	Оксид углерода	Диоксид азота	фенол	формальдегид	марганец
1	8	4,5	6,37	11	4,20	1,76
2	6	1,7	4,68	11	3,51	0,40
3	8	4,3	1,69	5	3,81	2,33
4	4	3,1	7,84	5	3,51	2,33
5	6	5,4	0,77	5	3,47	1,75
6	6	5,4	0,77	2	3,47	2,15
7	8	5,0	1,69	5	4,89	2,17
8	8	5,0	1,69	8	3,28	1,59
9	10	5,8	1,27	11	3,51	1,40
10	6	5,0	0,77	10	3,82	2,23
11	6	5,0	1,69	11	3,28	1,97
12	4	5,0	1,68	11	4,89	2,34
13	4	5,0	3,03	8	4,20	1,61
14	4	5,0	1,69	5	3,47	2,34
15	4	5,0	4,68	2	3,28	0,50
16	4	5,0	1,69	2	3,28	1,40
17	6	5,0	1,69	5	3,70	0,97
18	3	6,4	1,69	11	3,81	1,19
19	2	6,5	1,69	2	3,28	1,40
20	2	6,6	1,60	2	3,47	0,25
21	4	6,5	3,03	5	4,89	0,74

Таблица 2 Загрязнения города 2 (фрагмент мониторинга, у.е.)

годы	Пыль	Оксид углерода	Диоксид азота	фенол	формальдегид	марганец
1	6	1,5	2,667	8	3,5	0,8
2	4	0,667	2,222	8	2,75	0,16
3	6	1,45	1,333	6	2,25	1,7
4	2	1,067	4,444	6	2,75	1,7
5	4	1,833	0,889	6	3,25	2,5
6	4	1,833	0,889	4	3,25	1,1
7	6	1,667	1,333	6	3,75	2,0
8	6	1,667	1,333	10	3,0	0,7
9	8	2	1,156	8	2,75	0,6
10	4	1,667	0,889	7,4	2,0	1,2
11	4	1,667	1,333	8	3,0	3,5
12	2	1,667	1,33	8	3,75	1,4
13	2	1,667	1,778	10	3,5	3,2
14	2	1,667	1,333	6	3,25	1,4
15	2	1,663	2,222	4	3,0	0,2
16	2	1,667	1,333	4	3,0	0,6
17	4	1,667	1,333	2	2,5	0,4
18	5	2,333	1,333	8	2,25	0,5
19	1,8	2,5	1,333	4	3,0	0,6
20	1,8	2,67	1,3	4	3,25	0,1
21	2	2,5	1,778	6	3,75	0,3

7. ПОСТРОЕНИЕ РЕШАЮЩИХ ДИАГНОСТИЧЕСКИХ ПРАВИЛ.

Цель работы: овладение навыками структурно-параметрической идентификации диагностических правил для автоматизированных систем поддержки принятия решения и систем управления терапевтическим процессом.

Краткие теоретические сведения.

Любой объект (процесс) с точки зрения диагностики подвергается анализу со стороны исследователя, который, как правило, априори знает, какие существенные характеристики ему следует регистрировать для решения диагностической задачи. То есть, в этом случае, исследователь уже владеет набором диагностических правил, которые либо опровергают, либо подтверждают выдвинутую им рабочую гипотезу о состоянии объекта. Так как о каждом состоянии объекта может выдвигаться различное количество гипотез, то, следовательно, диагностические правила каждой из них не должны в случае объединения поглощать друг друга, и, вообще говоря, должны иметь минимальное количество пересечений как по регистрируемым параметрам, так и по диапазонам их изменений.

В общем случае диагностическое правило имеет вид, например, продукции: если значение $P=P_0$, то состояние $S=S_0$.

$$P=F(S, t, dS), \qquad (1)$$

где S - состояние; t - время; dS - диагноз изменения характеристик состояния.

Если зависимость (1) достаточно хорошо идентифицирована (с заданной степенью точности или неопределенности), то нетрудно построить экспертно-диагностическую систему продукционного типа с указанием исследователю технологии реализации необходимой информации для достаточно достоверной диагностики гипотетического состояния.

Рассмотрим **логический механизм** синтеза правила (1).

<u>1 этап</u>. Организация мониторинга состояния заданной глубины и полноты.

2 этап. Выделение множества ортогональных и информативных признаков с точки зрения вариативности. То есть, с одной стороны, селектируем сильно коррелированные характеристики, с другой стороны, отбираем те из них, вариативность которых (отношение дисперсии к среднему значению) выше определенного порогового уровня (например 10%).

3 этап. Кодируем состояния (лучше в двоичном коде): с учителем - то есть исследователь знает состояния, без учителя - выполняется кластер-анализ и задаются состояния или вводится пороговый принцип. Таким образом, получаем значения «логической» функции $Y=(Y_{i1},Y_{i2},Y_{il})$. Если состояний не много, то рекомендуется применять унитарное кодирование с минимизацией Хеменгового расстояния соседних состояний.

4 этап. Кодируем значение признакового пространства, следующим образом (во всех случаях рекомендуется унитарный код). По каждому оставленному признаку выделяем определенный набор состояний, как попадание значения признака в определенный диапазон. Диапазон определяется либо:

1) Экспертом, исходя из его знаний и жизненного опыта.

2) Исследователем, по анализу частоты распределений значений и личного опыта. При достаточно небольшом количестве признаков анализ гистограммы рекомендуется проводить визуально, наблюдая все признаки одновременно (в концепции системный подход).

3) Автоматически (с применением ЭВМ) по следующему алгоритму.

Исследователь задает количество состояний по каждому признаку n_i (каждое из них кодируется, желательно в унитарном коде). Определяется медиана Mo и дисперсия G_o. Определяется удельное отклонение как $G_y = G_o/(n_{i-1})$. В качестве первого диапазона (состояния) выбирается величина внутри диапазона $M_o \pm G_o$. Все значения X_i попавшие в данный диапазон кодируются определенным состоянием S_o. Величина n_i декрементируется и повторяется описанный процесс над «оставшимися» данными. Так продолжается до тех пор, пока n_i не станет равно 0 и всем оставшимся значениям будет присвоено состояние S_n. Граничные

значения $M_o \pm G_{yo}$ либо включаются в одно из состояний, либо, что более оптимально, кодируются знаком переходной функции.

5 этап. Определяем функциональные зависимости между полученными булевыми функциями (парные и множественные) и парное Хеминговое расстояние. Те признаки, у которых это расстояние равно нулю, селектируются путем оставления одного из них с наибольшей вариативностью.

Явный вид логической зависимости между булевыми переменными X_k, k=1,m определяются следующим образом. На первом шаге проверяются условия независимости: поскольку каждая булева функция может иметь два значения истинности, то m булевых функций может образовывать 2^m комбинаций значений истинности. Согласно определению m-булевых функций независимы, если в совокупности при всех возможных значениях аргументов они могут принимать 2^m комбинаций значений истинности. Т.е., для проверки независимости необходимо вычислить их изображающие числа и проверить, образуют ли они полный набор чисел. Если да, то функции независимы, в противном случае - зависимы.

На втором шаге в базисе булевых функций выписывают в последовательные строки изображающие числа и определяют какие числа отсутствуют в наборе столбцов (повторяющиеся значения чисел считают один раз). Столбцы набора представляют собой комбинации значений истинности функций $X_i,..., X_m$, при которых соответствующие элементарные произведения составленные из $X_i,..., X_m$ истинны. Таким образом, если идентифицируется зависимость:

$$F(X_i,...,X_n)=1 \qquad (2),$$

то, следовательно, имеющиеся в наборе столбцы указывают номера тех колонок базиса в $(X_1,...,X_n)$, которые совпадают с номерами изображающего числа $\#F(X_1,...,X_m)$, на которых функция F истинна. Например, пусть задан протокол мониторинга трех логических функций:

X_1 11001010
X_2 10101100
X_3 11001100

Выпишем последовательно все столбцы в этом наборе изображающих чисел как строки и укажем справа их десятичные значения:

111=7, 101=5, 010=2, 000=0, 111=7, 110=6, 001=1, 000=0

Видно, что десятичные эквиваленты 3 и 4 отсутствуют, а это означает, что по отношению и в (X_1, X_2, X_3) изображающее число связи F(X_1, X_2, X_3) = 1 имеет вид #F(X_1, X_2 ,X_3) = 1.

Минимизируя полученную функцию, получаем:

$$\#F=\bar{X}1\bar{X}3+X2X3+X1\bar{X}2=1$$

Проверяем:

X1	X2	$\bar{X}3$	$\bar{X}1X3$	X2X3	$\bar{X}1X2$	F
1	1	1	0	1	0	1
1	0	1	0	0	1	1
0	1	0	1	0	0	1
0	0	0	1	0	0	1
1	1	1	0	1	0	1
0	1	1	0	1	0	1
1	0	0	0	0	1	1
0	0	0	1	0	0	1

Таким образом, определяется каким образом логические функции связаны между собой.

<u>6 этап</u>. Идентифицируем логические функции Y=F(X) - парная зависимость и/или Y=F({X}) (3) - множественная зависимость. Заметим, что возможен вариант отсутствия тех или иных функциональных зависимостей.

<u>7 этап</u>. Переходим от полученных булевских функций либо к продукционным диагностическим правилам, либо к схемотехническому решению идентификационного диагностического устройства. Однако, второй вариант менее устойчив и мобилен в случае достаточно быстрого изменения окружающей среды, приводящего к изменению в функционировании анализируемого объекта (системы, процесса), а, следовательно, и вида идентифицированных функций.

Как и во множественном регрессионном анализе, при синтезе зависимостей (3) для получения более строгого результата (минимизации пересечений понятий в диагностических,

классификационных правилах каждого состояния) рекомендуется руководствоваться правилом максимальной организации (независимости) факторного пространства. Для этого необходимо добиться максимальной независимости X между собой, т.е. в идеале не должно существовать функциональных зависимостей между X_i. Т.е., если на пятом этапе идентифицируется F(x)=1 , то необходимо изменить множество X: либо путем исключения переменных (по критерию вариативности), что чревато в общем случае, потерей информации; либо изменить кодирование вводимых сигналов путем уменьшения количества состояний и/или изменения (экспертным путем) диагностических классов состояний. При достаточно мощной вычислительной технике и сравнительно небольшом размере факторного пространства (до 100 признаков) эти проблемы могут быть решены переборным путем. В противном случае, следует применять методы целенаправленного случайного поиска.

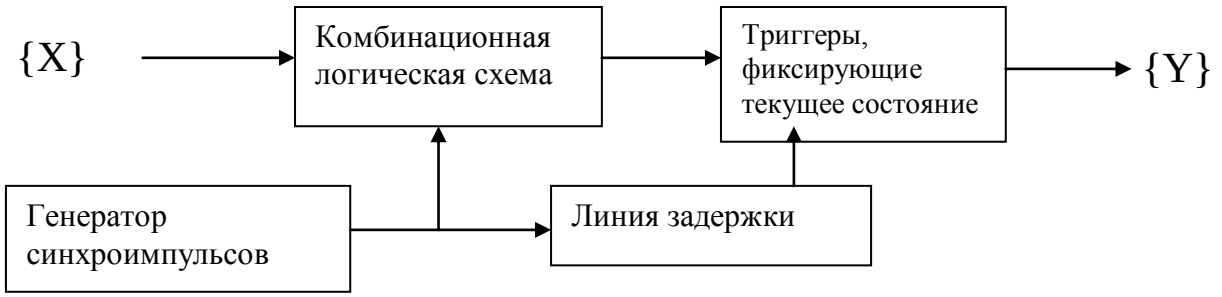
Порядок выполнения работы.

1. Согласно номеру варианта (равен порядковому номеру студента в журнале группы) N сформируйте протокол мониторинга наблюдения за процессом X, следующим образом:

Индекс переменной X	Индекс протокола Z (см. таблицу 1)
1	mod(n,8)+1
2	mod(n,8)+2
3	mod(n,8)+3
4	mod(n,8)+4

2. Выберите в качестве выходной величины, определяющей состояние процесса переменную мониторинга с индексом 4.
3. Зададитесь числом состоянием по Y- 3, X1-5, X2- 3, X3 -2.
4. Определите диапазон изменений состояний, причем для Y_1,X_1,- автоматически по дисперсии (здесь и далее рекомендуется использовать интегрированные среды типа EXCEL, STATISTICA), X_2 - экспертным путем анализа гистограммы, X_3 - экспериментальным путем заданием одного порога (<,>=).
5. Закодируйте состояние по X, У - т.е. получите характеристические числа (вид) булевых функций $X_б$ и $У_б$.

6. Проанализируйте взаимозависимость между булевскими факторами X и сформируйте наиболее ортогональное факторное пространство (векторы которого в наименьшей степени зависимы между собой).
7. Идентифицируйте в минимальном виде (с помощью карт Карно) логические функции $У_б=F(X_б)$.
8. Перейдите от булевского представления к логико-семантическому и сформулируйте диагностические правила продукционного типа.
9. Составьте схемотехническое решение диагностических правил по следующей структуре:



Рассчитайте скважность синхроимпульсов и их характер для обеспечения устойчивой работы схем, считая время срабатывания любого логического элемента 10мс, время срабатывания триггера 50мс.

11. Оформите отчет.

Контрольные вопросы:

1. Что определяет решающее правило?
2. Какие типы решающих правил применяют в диагностическом процессе при обработке результатов мониторинга?
3. В чем заключается логический способ синтеза решающего правила?
4. Каким образом осуществляется бинарное кодирование признакового пространства при синтезе логических решающих правил?
5. Каким образом реализуется схемотехническая реализация решающего правила на определенной электронной базе?

Библиография

1. Емельянов С.Г. Интеллектуальные системы на основе нечеткой логики и мягких арифметических операций [Текст] : учебник / С. Г. Емельянов , В. С. Титов, М. В. Бобырь. - Москва : Аргамак-Медиа, 2014. – 338с.
2. Кореневский Н. А. Проектирование систем поддержки принятия решений для медико-экологических приложений [Текст] : монография / КурскГТУ ; Курский государственный технический университет. - Курск : КурскГТУ, 2004. - 179 с.
3. Сазонов, Сергей Юрьевич (канд. тех. наук). Разработка подсистемы информационной поддержки принятия управленческих решений в медучреждении [Текст] : диссертация на соискание ученой степени кандидата технических наук : 05.13.10 / науч. рук. Уразбахтин И. Г. - Курск : [б. и.], 2003. – 167 с.
4. Оценка уровня функционального резерва организма в задачах прогнозирования и медицинской диагностики [Текст] : монография / В. И. Серебровский [и др.]. - Курск : КГСА, 2011. - 63 с.

Приложение

Таблица 1. Результаты регионального мониторинга

год	Всего родилось	Всего заболело	Врожденные пороки (ВП)	асфаксия	Умерло всего	Умерло от ВП	Умерло от асфиксии
1	1657	90	22	6	17	4	5
2	2081	170	24	9	32	3	9
3	2173	201	20	25	32	7	5
4	2676	198	41	17	34	5	5
5	2557	191	51	47	21	3	10
6	2522	586	83	78	23	1	3
7	2893	252	30	19	31	3	7
8	2956	270	45	8	32	6	5
9	2650	197	38	12	25	3	6
10	3036	213	42	36	36	4	12
11	3165	230	37	32	21	2	5
12	3181	218	61	42	27	8	3
13	2930	216	65	58	18	1	6
14	2491	202	41	55	20	2	5
15	2964	185	37	39	25	2	1
16	2425	290	65	87	22	2	7
17	2432	238	50	53	19	3	4
18	2388	196	34	65	19	0	9
19	2290	197	34	58	22	4	9
20	2995	193	45	27	28	5	7

Год	Ревм. пораж. сердца	Инфаркт миокарда	Гипертан. болезнь	стенокардия	Септический эндокартит	летальность
1	107	8	171	30	5	18
2	147	4	151	38	9	20
3	146	22	124	42	4	17
4	122	27	145	56	10	20
5	104	37	134	83	8	23
6	77	37	110	33	11	23
7	82	24	156	38	8	20
8	104	37	100	40	4	13
9	88	21	87	45	35	18
10	75	17	111	41	6	2
11	44	28	71	29	3	17
12	56	4	120	38	8	12
13	44	16	100	31	7	7
14	44	12	104	28	5	24
15	57	15	97	35	15	10
16	32	17	84	33	14	5
17	36	15	116	40	20	16
18	44	20	176	44	2	28
19	51	23	157	37	11	13
20	48	18	162	43	17	17

8. ПЛАНИРОВАНИЕ ЭКСПЕРИМЕНТА

Цель работы: овладение навыками планирования эксперимента при решении задач анализ медико-биологических систем с кибернетических позиций идентификации зависимости функции отклика от входных переменных «черного ящика».

Краткие теоретические сведения.

Планирование эксперимента (англ. experimental design techniques) - комплекс мероприятий, направленных на эффективную постановку опытов. Основная цель планирования эксперимента - достижение максимальной точности измерений при минимальном количестве проведенных опытов и сохранении статистической достоверности результатов.

Планирование эксперимента применяется при поиске оптимальных условий, построении интерполяционных формул, выборе значимых факторов, оценке и уточнении констант теоретических моделей и др.

Методы планирования эксперимента позволяют минимизировать число необходимых испытаний, установить рациональный порядок и условия проведения исследований в зависимости от их вида и требуемой точности результатов. Если же по каким-либо причинам число испытаний уже ограничено, то методы дают оценку точности, с которой в этом случае будут получены результаты. Методы учитывают случайный характер рассеяния свойств испытываемых объектов и характеристик используемого оборудования. Они базируются на методах теории вероятности и математической статистики.

Планирование эксперимента включает ряд этапов.

1. Установление цели эксперимента (определение характеристик, свойств и т. п.) и его вида (определительные, контрольные, сравнительные, исследовательские).

2. Уточнение условий проведения эксперимента (имеющееся или доступное оборудование, сроки работ, финансовые ресурсы, численность и кадровый состав работников и т. п.). Выбор вида испытаний (нормальные, ускоренные, сокращенные в условиях

лаборатории, на стенде, полигонные, натурные или эксплуатационные).

3. Выявление и выбор входных и выходных параметров на основе сбора и анализа предварительной (априорной) информации. Входные параметры (факторы) могут быть детерминированными, то есть регистрируемыми и управляемыми (зависимыми от наблюдателя), и случайными, то есть регистрируемыми, но неуправляемыми. Наряду с ними на состояние исследуемого объекта могут оказывать влияние нерегистрируемые и неуправляемые параметры, которые вносят систематическую или случайную погрешность в результаты измерений. Это — ошибки измерительного оборудования, изменение свойств исследуемого объекта в период эксперимента, например, из-за старения материала или его износа, воздействие персонала и т. д.

4. Установление потребной точности результатов измерений (выходных параметров), области возможного изменения входных параметров, уточнение видов воздействий. Выбирается вид образцов или исследуемых объектов, учитывая степень их соответствия реальному изделию по состоянию, устройству, форме, размерам и другим характеристикам.

На назначение степени точности влияют условия изготовления и эксплуатации объекта, при создании которого будут использоваться эти экспериментальные данные. Условия изготовления, то есть возможности производства, ограничивают наивысшую реально достижимую точность. Условия эксплуатации, то есть условия обеспечения нормальной работы объекта, определяют минимальные требования к точности. Точность экспериментальных данных также существенно зависит от объёма (числа) испытаний — чем испытаний больше, тем (при тех же условиях) выше достоверность результатов. Для ряда случаев (при небольшом числе факторов и известном законе их распределения) можно заранее рассчитать минимально необходимое число испытаний, проведение которых позволит получить результаты с требуемой точностью.

5. Составление плана и проведение эксперимента — количество и порядок испытаний, способ сбора, хранения и

документирования данных. Порядок проведения испытаний важен, если входные параметры (факторы) при исследовании одного и того же объекта в течение одного опыта принимают разные значения.

В ряде случаев, когда систематически действующие параметры сложно учесть и проконтролировать, их преобразуют в случайные, специально предусматривая случайный порядок проведения испытаний (рандомизация эксперимента). Это позволяет применять к анализу результатов методы математической теории статистики.

6. Статистическая обработка результатов эксперимента, построение математической модели поведения исследуемых характеристик. Необходимость обработки вызвана тем, что выборочный анализ отдельных данных, вне связи с остальными результатами, или же некорректная их обработка могут не только снизить ценность практических рекомендаций, но и привести к ошибочным выводам.

Обработка результатов включает:

- определение доверительного интервала среднего значения и дисперсии (или среднего квадратичного отклонения) величин выходных параметров (экспериментальных данных) для заданной статистической надежности;
- проверка на отсутствие ошибочных значений (выбросов), с целью исключения сомнительных результатов из дальнейшего анализа. Проводится на соответствие одному из специальных критериев, выбор которого зависит от закона распределения случайной величины и вида выброса;
- проверка соответствия опытных данных ранее априорно введенному закону распределения. В зависимости от этого подтверждаются выбранный план эксперимента и методы обработки результатов, уточняется выбор математической модели.

Построение математической модели выполняется в случаях, когда должны быть получены количественные характеристики взаимосвязанных входных и выходных исследуемых параметров. Это - задачи аппроксимации, то есть выбора математической зависимости, наилучшим образом соответствующей

экспериментальным данным. Для этих целей применяют регрессионные модели, которые основаны на разложении искомой функции в ряд с удержанием одного (линейная зависимость, линия регрессии) или нескольких (нелинейные зависимости) членов разложения (ряды Фурье, Тейлора). Одним из методов подбора линии регрессии является широко распространенный метод наименьших квадратов.

Для оценки степени взаимосвязанности факторов или выходных параметров проводят корреляционный анализ результатов испытаний. В качестве меры взаимосвязанности используют коэффициент корреляции: для независимых или нелинейно зависимых случайных величин он равен или близок к нулю, а его близость к единице свидетельствует о полной взаимосвязанности величин и наличии между ними линейной зависимости.

При обработке или использовании экспериментальных данных, представленных в табличном виде, возникает необходимость получения промежуточных значений. Для этого применяют методы линейной и нелинейной (полиноминальной) интерполяции (определение промежуточных значений) и экстраполяции (определение значений, лежащих вне интервала изменения данных).

7. Объяснение полученных результатов и формулирование рекомендаций по их использованию, уточнению методики проведения эксперимента.

Снижение трудоемкости и сокращение сроков испытаний достигается применением автоматизированных экспериментальных комплексов. Такой комплекс включает испытательные стенды с автоматизированной установкой режимов (позволяет имитировать реальные режимы работы), автоматически обрабатывает результаты, ведет статистический анализ и документирует исследования. Но велика и ответственность инженера в этих исследованиях: четкое поставленные цели испытаний и правильно принятое решение позволяют точно найти слабое место изделия, сократить затраты на доводку и итерационность процесса проектирования.

Стандартные планы. Многие планы экспериментов в настоящее время стандартизованы. Они имеются в справочниках, математических пакетах программ и системах моделирования. Однако исследователь должен быть готов к модификации имеющихся планов и приспособлению их к специфическим условиям конкретных задач. С полным факторным экспериментом мы уже знакомы. Это, как отмечалось ранее, самый информативный план, понятный по структуре, но и самый неэкономичный. Поэтому ПФЭ применяют, когда число факторов невелико.

Пример: Необходимо провести эксперимент с моделью, имеющей три двухуровневых фактора, с целью построения математической модели ("вторичной модели") процесса в виде:
$y = b_0+b_1x_1+b_2x_2+b_3x_3+b_{12}x_1x_2+b_{13}x_1x_3+b_{23}x_2x_3+b_{123}x_1x_2x_3.$
Уравнение имеет восемь коэффициентов, следовательно, достаточно провести восемь наблюдений. Это уравнение соответствует ПФЭ типа $N = 2^3 = 8$.

Полный факторный эксперимент дает возможность определить не только коэффициенты b_1, b_2, b_3, соответствующие так называемым линейным эффектам (их также называют главными), но и коэффициенты $b_{12}, b_{13}, b_{23}, b_{123}$, соответствующие всем эффектам взаимодействия факторов, а также свободный член b_0. Эффекты взаимодействия двух и более факторов проявляются, если влияние каждого из них на отклик зависит от уровней, на которых установлены другие факторы.

Теперь допустим, что число наблюдений в эксперименте, равное восьми, неприемлемо и план надо сократить. Вполне естественно предположить, что эффекты взаимодействия оказывают на реакцию системы существенно меньшее влияние, чем линейные, или даже отсутствуют вовсе, если факторы обладают свойством независимости. Исключим их и тогда модель процесса (уравнение отклика, уравнение реакции, "вторичная модель") принимает вид: $y = b_0 + b_1x_1 + b_2x_2 + b_3x_3.$

Теперь число неизвестных коэффициентов b_i сократилось вдвое и число необходимых наблюдений для их определения стало равно четырем.

Что это за наблюдения?

Четыре наблюдения достаточны для проведения ПФЭ при двух факторной модели $N = 2$. Этими факторами, например, могут быть x_1, x_2 или другая двухфакторная комбинация из трех факторов. Уровни третьего фактора x_3 получают из первых двух с помощью, так называемого генерирующего соотношения:

Поскольку факторы двухуровневые, то в общем виде уровни принято обозначать так: верхний уровень: +1; нижний уровень: -1. Новый, сокращенный план эксперимента называют полурепликой и обозначают 2^{3-1}. Таким же образом можно проводить дальнейшее сокращение планов типа 2^{k-1}, получая четверть реплики 2^{k-2} и более мелкие реплики. Естественно, такое сокращение числа экспериментов приводит к "огрублению" коэффициентов b_i. Следовательно, полученную модель процесса $y = f(x)$ нужно проверять на адекватность, используя для этого "сэкономленные" наблюдения.

Рассмотренное планирование является основой и составной частью для разработки более сложных - несимметричных многоуровневых планов.

Не менее часто целью экспериментов является проверка разного рода гипотез о природе сравниваемых объектов. Например, однородны ли выходы двух систем в смысле законов распределения, характеристик этих законов. Поскольку обработка данных эксперимента ведется методами дисперсионного анализа, то и планы в данном случае называются планами дисперсионного анализа. Сущность дисперсионного анализа мы рассмотрим в следующей теме.

Планы дисперсионного анализа могут быть полные, если используются все возможные сочетания условий (аналогично ПФЭ), и неполные, которые применяются тогда, когда полные планы оказываются громоздкими и неэкономичными. Сокращение планов происходит, как и ранее, за счет исключения некоторых

сочетаний факторов (взаимодействий) и уровней случайным или традиционным образом.

Наиболее популярными из неполных планов является симметричный план "латинский квадрат" или его вариации. Этот план целесообразно применять, когда из всех существенных факторов можно выделить один доминирующий (самый существенный).

В планах дисперсионного анализа часто факторы обозначают латинскими буквами $A, B, C, \ldots$, а уровни - индексами при соответствующих факторах: A_1, A_2, B_1.

Пример. Требуется построить план "латинский квадрат" симметричного трехфакторного четырехуровневого эксперимента. Доминирующий фактор A.

Решение

Исходные данные: $k = 3, q = 4$.

Введем обозначения факторов и уровней:

A_1, A_2, A_3, A_4 - уровни доминирующего фактора A ;

B_1, B_2, B_3, B_4 - уровни фактора B ;

C_1, C_2, C_3, C_4 - уровни фактора C.

План приведен в таблице

План "латинский квадрат"				
Уровни B	Уровни C			
	C_1	C_2	C_3	C_3
B_1	A_1	A_2	A_3	A_4
B_2	A_2	A_3	A_4	A_1
B_3	A_3	A_4	A_1	A_2
B_4	A_4	A_1	A_2	A_3

В этом плане число наблюдений $N = 4 \cdot 4 = 16$. В полном плане их было бы $N = 4^3 = 64$. Сокращение произошло за счет исключения некоторых комбинаций: $A_1B_2C_1$, $A_1B_2C_2$ и др. Заметим, что план может быть и несимметричным. В этом случае вместо квадрата будет прямоугольник. И еще: выделение доминирующего фактора не является существенным, то есть, внутри квадрата можно располагать уровни любого из действующих факторов.

В практике планирования экспериментов встречаются и такие неполные планы: один из факторов меняет свои значения при фиксированных значениях других. То есть исследуется поочередно влияние каждого фактора в отдельности.

Иногда применяются и так называемые рандомизированные планы. В таких планах сочетания факторов и уровней для каждого прогона модели выбираются случайно. Вид случайности и объем выборки определяется исследователем.

Формальный подход к сокращению общего числа прогонов

Рассмотренные способы сокращения общего числа прогонов носят эвристический (субъективный) характер. Они осуществлялись за счет исключения каких-то комбинаций уровней факторов.

Однако во многих случаях исследователь имеет свободу действий в выборе числа факторов k, числа уровней q и числа прогонов p модели в одном наблюдении. Каждый из этих аргументов в конкретной ситуации по-разному влияет на общее число прогонов модели N.

Исследуем эти влияния. Как нам уже известно, общее число прогонов (реализаций) модели равно: $N = pq^k$.

Рассмотрим относительное влияние аргументов p, q, k на число реализаций N.

Сначала нужно получить выражения для вычисления скоростей изменения функции N при изменении одного аргумента и неизменных остальных аргументах. Для этого последовательно найдем частные производные первого порядка от функции N по этим аргументам:

$$\frac{\partial N}{\partial k} = pq^k \ln q$$

$$\frac{\partial N}{\partial q} = pkq^{k-1}$$

$$\frac{\partial N}{\partial p} = q^k$$

Теперь сравним попарно полученные производные:

$$\frac{\partial N}{\partial k} \Big/ \frac{\partial N}{\partial q} = \frac{q \ln q}{k}$$

$$\frac{\partial N}{\partial p} \Big/ \frac{\partial N}{\partial q} = \frac{q}{kp}$$

$$\frac{\partial N}{\partial p} \Big/ \frac{\partial N}{\partial k} = \frac{1}{p \ln q}$$

Из соотношений 1 и 2 следует: если $(kp \succ q$ и $(k \succ q \ln q)$, то наибольшее влияние на число N оказывает изменение числа уровней q. Из соотношений 3 и 1 следует: если $(p \ln q \succ 1)$ и $(k \prec q \ln q)$, то наибольшее влияние на число N оказывает изменение числа факторов k. Из соотношений 2 и 3 следует: если $(q \succ kp)$ и $(1 \succ p \ln q)$, то наибольшее влияние на число N оказывает изменение числа реализаций модели на каждом уровне факторов (на каждом наблюдении).

Рассмотренный формальный подход к сокращению числа реализаций не совсем корректен, так как функция общего числа прогонов N носит не непрерывный, а дискретный характер. Тем не менее, такой подход применяется с последующим округлением результатов до целых чисел.

Покажем применение формального подхода сокращения реализаций на примере.

Пример. На вход модели объекта действуют четыре трехуровневых фактора $(k = 4, q = 3)$. В каждом наблюдении предполагаются восемь прогонов модели $(p = 8)$. Полный факторный эксперимент потребует $N = pq^k = 8 \cdot 3^4 = 648$ прогонов или 81 наблюдение. Такие затраты ресурсов неприемлемы. Требуется определить, какой из аргументов (k, q, p) следует уменьшить, чтобы достичь наиболее существенного уменьшения числа реализаций N.

Решение

Подготовим данные для сравнений:

$q \ln q = 3 \ln 3 = 3.3;\ kp = 4 \cdot 8 = 32;\ p \ln q = 8 \cdot 1.1 = 8.8.$

Соблюдается условие:

$(kp \succ q) \wedge (k \succ q \ln q)$, так как $(32 \succ 3) \wedge (4 \succ 3.3)$.

Следовательно, наибольшее влияние на изменение N оказывает изменение числа уровней q.

Уменьшим q на единицу: $q = 3 - 1 = 2$. В этом случае при ПФЭ потребуется выполнить $N = 8 \cdot 2 = 128$прогонов или 16 наблюдений, то есть в пять раз меньше.

Варьирование факторов на двух уровнях встречается часто и решение $q = 2$ будет приемлемо, если нет обстоятельств, не устраивающих это решение.

Элементы тактического планирования

Основной задачей тактического планирования является обеспечение результатам компьютерного эксперимента заданных точности и достоверности. Рассмотрим случай, когда имитационная модель строилась для определения характеристик некоторых случайных величин. Такими случайными величинами могут быть:

- время обслуживания заявки в СМО;
- численности противоборствующих сторон;
- расход боеприпасов;
- время наработки на отказ технического устройства и др.

Из характеристик случайных величин, как правило, интересуют среднее значение (матожидание), дисперсия и характеристика связи случайных величин - коэффициент корреляции. Характеристику случайной величины будем обозначать греческой буквой Θ. С помощью имитационного моделирования точное значение Θ определить нельзя, так как число N реализаций модели конечно.

При конечном числе реализаций модели определяется приближенное значение характеристики. Обозначим это приближение $\overline{\Theta}$.

Приближенное значение $\overline{\Theta}$ называют оценка соответствующей характеристики: оценкой матожидания, оценкой дисперсии, оценкой коэффициента корреляции.

Точностью характеристики $\overline{\Theta}$ называют величину ε в отношении $|\overline{\Theta} - M[\Theta]| < \varepsilon,$ где $M[\Theta]$ - матожидание случайной величины.

Величина ε представляет собой абсолютное значение ошибки в определении значения искомой характеристики.

Достоверностью оценки характеристики $\overline{\Theta}$ называют вероятность α того, что заданная точность достигается: $P(|\overline{\Theta} - M[\Theta]| < \varepsilon) = \alpha.$ Достоверность характеризует повторяемость, устойчивость эксперимента и трактуется так: если для оценки $M[\Theta]$ использовать величину $\overline{\Theta},$ то в среднем на каждые 1000 применений этого правила в $1000 \cdot \alpha$ случаев величина $\overline{\Theta}$ будет отличаться от $M[\Theta]$ на величину меньше ε.

В ряде случаев целесообразно пользоваться понятием относительной точности $d = \frac{\varepsilon}{M[\Theta]}$. В этом случае достоверность оценки имеет вид:

$$P\left(\frac{\overline{\Theta} - M[\Theta]}{M[\Theta]} < d\right) = \alpha$$

Структура медико-биологического эксперимента.

Объектом исследований в медико-биологических науках являются живые организмы, как правило – в срезе их взаимодействия с комплексом различных факторов внешней среды, часть из которых относится к болезнетворным. Другие факторы направлены на профилактику или лечение болезни. Основным подходом в этом направлении по-прежнему остается эксперимент, а основной его целью – установление закономерностей возникновения различных болезней, механизмов их развития, разработка и проверка эффективности новых методов профилактики и лечения.

В процессе познания в биологии и медицине научный эксперимент выполняет следующие основные функции:

- выступает средством получения новых научных данных;

- является способом выделения общего в серии сходных явлений, обоснования закономерностей, формирования гипотез;
- выступает средством проверки гипотез и теорий, критерием их истинности, основой для выдвижения новых гипотез;
- является относительным гарантом перед применением новых методов диагностики и лечения болезней в клинической практике.

По своей продолжительности различают острый эксперимент (например, воспроизведение острой кровопотери, ожоговой травмы, отравления угарным газом и т.д.) и хронический, который подразумевает длительное наблюдение за экспериментальным объектом (например, животным: введение электродов, операция изолированного желудка по И.П. Павлову, трансплантация органов и тканей и т.д.).

Эксперимент позволяет производить такие воздействия, например, на органы животного (травма, отравление, облучение), которые невозможны и недопустимы в клинике. В эксперименте на животных возможно испытание новых лекарственных препаратов. Некоторые болезни (например, психические заболевания, ряд опухолей, некоторые иммунопатологические состояния) пока в эксперименте воспроизвести не удалось.

Этапами постановки и проведения эксперимента являются: формирование рабочей гипотезы, определение цели и задач исследования, выбор конкретных методик, непосредственное проведение эксперимента (серии опытов), фиксация и анализ данных эксперимента, обсуждение и выводы.

Рабочая гипотеза – первое, с чем сталкивается экспериментатор. Это – попытка объяснить какой-либо процесс на основе уровня знаний, имеющегося в настоящий момент. Нельзя просто ставить опыт, не зная, какие предположительно изменения в состоянии объекта в ту или другую сторону можно зафиксировать.

Основу проведения эксперимента составляет получение максимально точной (адекватной) модели болезни. Моделирование – это воспроизведение структурно-функционального комплекса болезни в более упрощенной форме для выяснения причин, условий и механизмов ее развития, разработки методов лечения и профилактики.

Требования к модели:
- аналогия причин и условий, вызывающих болезнь и примененных при ее моделировании;
- сходство функциональных и морфологических изменений болезни и ее модели на системном, органном, клеточном и молекулярном уровнях;
- сходство в развитии типичных осложнений;
- эффективность лечения одними и теми же воздействиями.

Например, в патологии, в зависимости от степени упрощения и уровня изучения, можно моделировать: типовые патологические процессы, отдельные нозологические формы болезни, нарушения отдельных органов и систем, нарушения деятельности отдельных клеток и их компонентов, молекулярные нарушения.

Главный вопрос, стоящий перед исследователем – это правомерность экстраполяции данных, полученных экспериментальным путем на животных на человеческий организм. Можно говорить только об условной адекватности используемых моделей, так как полная идентичность проявлений заболеваний у животных и у человека принципиально недостижима. Этому препятствуют иные законы, царящие в человеческом обществе.

В человеческом обществе в определенной мере отсутствует естественный отбор, что предполагает наличие определенного числа вырождающихся функций и приобретения патологических реакций. Помимо этого, чрезвычайно важную роль в формировании патологии у человека играет социальный строй, адекватное моделирование которого практически невозможно. Связанным с социальной функцией является и наличие второй сигнальной системы, играющей колоссальную роль в процессах компенсации и адаптации. Неслучайно еще со времен Гиппократа было замечено, что у врача есть три метода воздействия на больного, причем на первом месте, еще до «травы» и «ножа», стоит «слово». Кроме того, некоторые заболевания человека, такие, как корь и скарлатина, вообще невозможно воспроизвести на животных ввиду отсутствия у них того молекулярного субстрата, на котором данная болезнь может существовать.

Выполнение экспериментальных исследований строго регламентировано законодательством. Все работы проводятся только в рамках утвержденных учебных и научных программ, специалистами с медико-биологическим образованием, болезненные и травмирующие процедуры выполняются с необходимым уровнем обезболивания. Все эксперименты проводятся не из праздного любопытства, а для изучения механизмов развития заболеваний, актуальных для человека, это первый, необходимый, но не самодостаточный этап познания болезни.

Рассмотрим следующую *модель медико-биологического эксперимента.* Пусть имеется некоторый объект, изменение состояния которого исследуется в ходе эксперимента. В качестве объекта может выступать отдельный организм, группа лабораторных животных (нередко в литературе – серия) и т.д. Состояние объекта измеряется теми или иными показателями (характеристиками) по критериям, отражающим его существенные характеристики. Примерами критериев являются: выраженность интоксикации, выживаемость в группе животных на определенный период после начала опыта и т.д., примерами характеристик – температура, активность тех или иных ферментов в биологических жидкостях, количественные показатели структуры внутренних органов и т.д. При этом мы по умолчанию подразумеваем, что методы измерения характеристик объектов одинаковы.

Эксперимент заключается в целенаправленном воздействии на объект, призванном изменить его определенным образом. Собственно, это воздействие – его состав, структура, свойства и т.д. – и есть результат теоретического (теоретической части) исследования. Примерами воздействия являются любые методы воздействия на болезнь с целью ее излечения, включая потенциальные лекарственные средства, немедикаментозные воздействия, хирургические манипуляции.

Следовательно, при проведении медико-биологического эксперимента необходимо обосновать, что состояние объекта изменилось, причем в требуемую сторону. Но этого оказывается

недостаточно. Ведь нужно доказать, что изменения произошли именно в результате произведенного воздействия.

Действительно, например, на утверждение о том, что температура тела у экспериментального животного снизилась в результате использования нового испытуемого вещества, можно всегда возразить, – а, может быть, она снизилась бы сама, без каких-либо нововведений, или в результате каких-либо других воздействий? Аналогично, на утверждение о том, что скорость и степень снижения температуры у животных, которым вводился новый препарат, отличаются от того, как это происходило у животных, леченных с применением традиционных препаратов, можно возразить, – а, может быть, сама группа имела до начала эксперимента внутренние отличия, позволившие ей продемонстрировать подобные «успешные» результаты.

Таким образом, для того, чтобы выделить в явном виде результат целенаправленного воздействия на исследуемый объект, необходимо взять аналогичный объект и посмотреть, что происходит с ним в отсутствии воздействий.

Традиционно эти два объекта называют соответственно экспериментальной группой (иногда основной) и контрольной группой (или группой сравнения).

На рисунке. 1 представлена в общем виде структура эксперимента (двойными пунктирными стрелками отмечены процедуры сравнения характеристик объектов).

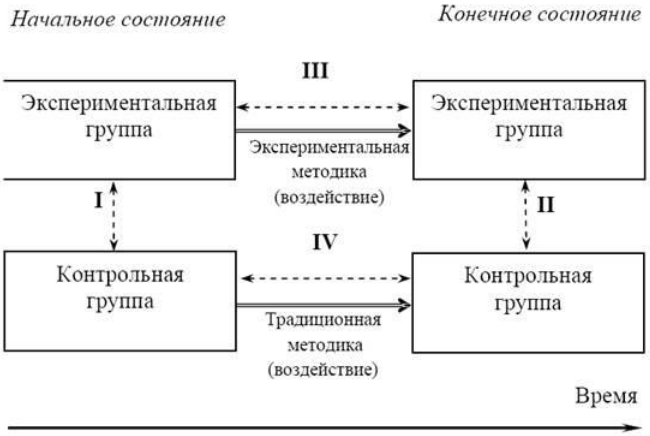


Рис. 1. Структура эксперимента

Констатации (в результате сравнения III – см. Рис. 1) различий начального и конечного состояний (динамики) экспериментальной группы недостаточно – быть может, аналогичные изменения происходят и с контрольной группой, что может быть установлено сравнением IV. Поэтому алгоритмдействий исследователя заключается в следующем:

1) На основании сравнения I установить совпадение начальных состояний экспериментальной и контрольной группы. Если говорить корректно, то с точки зрения математической статистики совпадение установить невозможно – можно установить различие или отсутствие статистически значимого различия. Точнее, при статистической проверке гипотез по статистическим данным выбирают одну из двух гипотез – есть различие или нет различия (точнее, различие не обнаружено);

2) Реализовать воздействие на экспериментальную группу. При выполнении данного шага необходимо быть уверенным, что и экспериментальная, и контрольная группы находятся в одинаковых условиях, за исключением целенаправленно изменяемых исследователем;

3) На основании сравнения II установить различие конечных состояний экспериментальной и контрольной группы.

После выполнения трех перечисленных шагов можно приступать к изучению зависимостей между различными характеристиками объектов.

Легко видеть, что, выполняя перечисленные шаги, мы, фактически, косвенным образом реализуем процедуру сравнения III, исключая влияние общих для экспериментальной и контрольной группы условий и воздействий.

Задачи анализа данных

Возможно выделение следующих типовых (для рассматриваемой предметной области) задач анализа данных.

ЭТАП	*Описание данных*	*Изучение сходства/различий*	*Исследование зависимостей*	*Снижение размерности*	*Классификация и прогноз*
МЕТОДЫ	- описательная статистика. - определение необходимого объема выборки.	Статистические критерии: Крамера-Уэлча. Вилкоксона-Манна-Уитни. χ^2. Фишера.	- корреляционный анализ. - дисперсионный анализ. - регрессионный анализ.	- факторный анализ. - метод главных компонент.	- дискриминантный анализ. - кластерный анализ. - группировка.

1. *Описание данных* - компактное описание имеющихся данных с помощью различных показателей и графиков. К этому классу можно отнести также задачу определения необходимого объема выборки.

2. *Изучение сходства/различий* (сравнение двух выборок). Задача заключается в установлении совпадений или различий характеристик двух выборок. В соответствии с описанной выше схемой эксперимента (см. Рис. 1) используются две группы – экспериментальная и контрольная. При этом считается, что каждая выборка является результатом независимых наблюдений реализации некоторой случайной величины с соответствующей функцией распределения, неизвестной исследователю.

Предполагается также, что наблюдения в одной выборке не зависят от наблюдений в другой, поэтому выборки называются независимыми. Возможен другой вариант – когда выборки связаны, например, рассматриваются одни и те же пациенты до и после лечения. Предположим, что для каждого пациента имелся небольшой положительный сдвиг. Если рассматривать результаты измерений в начале и в конце лечения как независимые выборки и применять соответствующие методы проверки однородности, то эффект лечения может не быть обнаружен, поскольку будет скрыт большим размахом данных. И только методы, основанные на анализе связанных выборок, обнаружат эффект (например, критерий знаков).

3. *Исследование зависимостей*. Следующим шагом после изучения сходства/различий является установление факта наличия/отсутствия зависимости между показателями (переменными - будем употреблять эти термины как синонимы) и

количественное описание этих зависимостей. Для этих целей используются, соответственно, корреляционный и дисперсионный анализ, а также - регрессионный анализ.

4. Снижение размерности заключается в уменьшении числа анализируемых переменных либо посредством выделения существенных переменных, либо/и построения новых показателей (на основании полученных в результате эксперимента). Соответствуют статистические методы: факторный анализ, метод главных компонент. Они не рассматриваются в курсе.

5. *Классификация и прогноз.*

Объектом изучения медико-биологической статистики являются совокупности. *Совокупностью* называют множество сходных, в некотором смысле однородных объектов, на которых производят идентичные измерения (наблюдения). Например, совокупность детей какого-либо района в возрасте до 3-х лет, совокупность животных определенного вида или совокупность животных, предназначенных для эксперимента, совокупность студентов в этой аудитории и т.д. Совокупность состоит из единиц совокупности или членов. Количество единиц (членов), составляющих совокупность, называют объемом совокупности и обозначают n.

Наиболее общую совокупность всех членов, которые могут быть к ней отнесены (даже мысленно), называют генеральной. В теоретических рассуждениях n генеральной совокупности стремится к бесконечности. В генеральных совокупностях, используемых в биологии и медицине, количество единиц, как правило, настолько велико, что оценить состояние всех ее членов практически невозможно (или нецелесообразно), поэтому объектами изучения являются только части генеральной совокупности, которые называют выборочными совокупностями. Выборочной совокупностью или просто выборкой называют отобранную для исследования часть генеральной совокупности.

В медицине оценку состояния членов совокупности проводят с помощью некоторых признаков: антропометрических (рост, вес,

объем груди), физиологических (частота дыхания, частота сердечных сокращений, давление крови, ЭКГ, ЭЭГ), биохимических (активность ферментов, концентрации гормонов в сыворотке крови, метаболитов в мочи, слюне, кале), морфологических (формула крови, гистология биопсийного материала). Следует отметить, что ранее методы статистики применялись лишь в научных исследованиях. Однако в настоящее время статистические исследования широко используются в условиях клиник, оснащенных компьютерами, даже для обработки данных одного конкретного пациента. Эти исследования позволяют значительно повысить точность диагноза.

Значения признаков при измерениях могут изменяться, варьировать, поэтому их называют вариантами и обозначают, как правило, маленькими буквами латинского алфавита с цифровыми индексами: x_1, x_2, x_3, …, x_n. Для обозначения любой варианты ряда вместо цифры используют индекс i (i = 1, 2, 3, …, n).

Если некоторое i-ое значение варианты при измерениях встретилось n_i раз, то n_i называют частотой. Таблицу, отображающую соответствие между значениями вариант (x_i) и их частотами (ni) называют вариационным рядом.

Если варианты расположены по возрастающей, то ряд называют ранжированным.

Ясно, что объем выборки n равен сумме частот $n_1+n_2+...+n_k$, где k - число различных вариант в вариационном ряду. Величину $\frac{n_i}{\sum n_i}$ называют относительной частотой.

Показатели описательной статистики можно разбить на несколько групп:

- показатели положения описывают положение экспериментальных данных на числовой оси. Примеры таких данных – максимальный и минимальный элементы выборки, среднее значение, медиана, мода и др.;
- показатели разброса описывают степень разброса данных относительно своего центра (среднего значения). К ним относятся: выборочная дисперсия, разность между минимальным и максимальным элементами (размах, интервал выборки) и др.

- показатели асимметрии: положение медианы относительно среднего и др.
- гистограмма и др.

Если члены ряда возрастают в геометрической прогрессии, то средняя арифметическая плохо характеризует среднюю тенденцию ряда. В этом случае усредняют произведение вариант

$\bar{x}_g = \sqrt[n]{x_1 \cdot x_2 \cdot x_3 \cdot \ldots \cdot x_n} = \sqrt[n]{\prod_{i=1}^{n} x_i}$ $\bar{x}_g$ называют средней геометрической. Среднюю геометрическую используют при изучении темпов роста организмов или роста целых популяций.

Количественные характеристики вариационных рядов, вычисленные по результатам измерений на выборочной совокупности (выборочные характеристики), рассматриваются в математической статистике как приближенные или точечные оценки соответствующих параметров генеральной совокупности, которые, как правило, остаются неизвестными.

Так выборочная средняя ($\bar{x}$) является точечной оценкой генеральной средней (m), выборочная дисперсия (S2) служит оценкой генеральной дисперсии (σ^2), среднее квадратичное отклонение (S) - точечная оценка стандартного отклонения (σ) генеральной совокупности, объем которой стремится к бесконечности. Как правило, точечные оценки не совпадают с соответствующими генеральными параметрами. Величину отклонения выборочного показателя от соответствующего генерального параметра характеризуют с помощью ошибки репрезентативности.

Ошибка репрезентативности обусловлена случайным отбором членов выборки из генеральной совокупности и не является ошибкой измерений, возникающей при измерениях показателей жизнедеятельности биологических объектов.

Ошибку репрезентативности средней арифметической (ошибку средней) определяют по формуле:

$$S_{\bar{x}} = \sqrt{\frac{(x_1 - \bar{x})^2 + (x_2 - \bar{x})^2 + \ldots + (x_n - \bar{x})^2}{n(n-1)}}$$. В медицине ошибку средней часто обозначают m.

Порядок выполнения работы.

1. Считая организм человека автономной системой управления реагирующей на различные воздействия спланируйте, проведите и обработайте результаты эксперимента по изучению реакции различных анализаторов и органов чувств на внешние раздражители. Вид эксперимента выбирается из следующих, согласно варианту, определенному преподавателем:

- Вариант 1. Проведите исследование костной и воздушной проводимости звука.
- Вариант 2. Проведите исследование тактильной чувствительности по описанной в Приложении сведениях методике.
- Вариант 3. Проведите исследование температурной чувствительности по описанной в приложении методике.
- Вариант 4. Определите порог вкусовой чувствительности по описанной в приложении методике.
- Вариант 5. Определите остроту обоняния по описанной в теоретических сведениях методике.

2. Оформите отчет.

Контрольные вопросы

1.Определение эксперимента.
2. Для чего предназначен эксперимент?
3. Определение опыта.
4. Что такое активный и пассивный эксперименты?
5. Определение плана эксперимента.
6. Какие факторы задаются в плане эксперимента?
9. Что такое планирование эксперимента?
10. В чем состоит принцип отказа от полного перебора?
11. В чем состоит принцип последовательного планирования?
12. В чем состоит принцип сопоставления с шумом?
13. В чем состоит принцип рандомизации?
14. В чем состоит принцип оптимальности плана?

15. Каким условиям должна удовлетворять информация, полученная в результате правильно спланированного эксперимента?
16. Как можно управлять эффективностью экспериментальных оценок?
17. Общий вид латинских квадратов.
18. Планирование объема эксперимента.
19. Использование среднего квадратического отклонения для планирования объема эксперимента.
20. Использование доверительного интервала для планирования объема эксперимента.
21. Использование статистических критериев для планирования объема эксперимента.
22. Суть однофакторного эксперимента.
23. Типовая гипотеза однофакторного эксперимента.
24. Вид дисперсионной математической модели однофакторного эксперимента.
25. На какие составляющие разбивается дисперсия результатов однофакторного эксперимента?
26. Чем оценивается значимость исследуемого фактора?
27. Что такое полный факторный эксперимент?
28. Что такое полный план?
29. Суть двухфакторного эксперимента.
30. Типовая гипотеза двухфакторного эксперимента.
31. Понятие полных и неполных блоков плана.
32. Что такое сбалансированные блоки?
33. Типовая гипотеза трехфакторного эксперимента.
34. План линейного трехфакторного двухуровневого эксперимента.
35. Что такое симметричность плана?
36. Что такое условие нормировки плана?
37. Что такое ортогональность плана?
38. Что такое ненасыщенный и сверхнасыщенный планы?
39. Правило для построения полного плана.
40. Связь полноты плана с моделью плана.
41. Каким условиям должен удовлетворять дробный план?
42. Правило для построения дробного плана.

43. На основе какого свойства факторов факторный анализ позволяет выбрать исследуемые факторы?

Библиография

1. Дрейзин В. Э. Основы научных исследований и инженерного творчества [Электронный ресурс]: учебное пособие / В. Э. Дрейзин; И. С. Захаров. - Курск : КурскГТУ, 2005 - .Кн. 2: Математическая обработка экспериментальных данных и построение по ним математических моделей объектов / Министерство образования Российской Федерации, Курский государственный технический университет. - 173 с.
2. Методы планирования эксперимента / http://www.studfiles.ru/preview/1560712/
3. Об утверждении Правил проведения медико-биологических экспериментов, доклинических (неклинических) и клинических исследований /http://adilet.zan.kz/rus/docs/V090005932_
4. Современные методы медико-биологических исследований./ http://sud-expertiza.ru/sovremennye-metody-mediko-biologicheskih-issledovaniy/
5. Соловьев В.П. Организация эксперимента: учебное пособие / В. П. Соловьев, Е. М. Богатов. -Старый Оскол: ТНТ, 2013. – 256с.
6. Структура медико-биологического эксперимента / http://lektsii.org/3-78861.html
7. Статистические методы. Планирование экспериментов: термины и определения. - Изд. офиц. - М.: Госстандарт России, 2021. - 35 с.

Приложение

КОСТНАЯ И ВОЗДУШНАЯ ПРОВОДИМОСТЬ ЗВУКА.

В классическом виде исследования проводятся с помощью камертона. В данном случае предлагается использовать имеющиеся у учащихся средства мобильной связи и наушники как источник звука. Источник звука прикладывается: к темени, затылку, зубам, шеи, тыльной стороны руки, к каждому из ушей (второе закрывается тампоном) (как внутри уха, так и в 1 см от ушной раковины непосредственно около ушного отверстия). Фиксируется сила звука (громкость), при которой воспринимается и

идентифицируется звуковой сигнал (например, слова песни или определенная мелодия).

ИССЛЕДОВАНИЕ ТАКТИЛЬНОЙ ЧУВСТВИТЕЛЬНОСТИ.

В коже рецептора находятся рецепторы тактильной, болевой и температурной чувствительности. Тактильные рецепторы расположены на поверхности тела неравномерно (больше всего их на кончиках пальцев, ладонях, кончике язык, меньше всего – на спине).

Оценка тактильной чувствительности осуществляется следующим образом. Испытуемого, сидящего на стуле, просят закрыть глаза. Циркулем одновременно прикасаются к различным участкам тела (пальцы рук, ладони, рука) и фиксируют на каком расстоянии между ножек циркуля различаются обе.

ИССЛЕДОВАНИЕ ТЕМПЕРАТУРНОЙ ЧУВСТВИТЕЛЬНОСТИ (ТЕРМОЭСТЕЗИОМЕТРИЯ).

Холодовые терморецепторы располагаются в поверхностных слоях кожи (0,16 мм), тепловые – в более глубоких слоях (около 0,3 мм). Распределение терморецепторов в коже неравномерно. Меньше всего их в коже лица, больше всего – в коже конечностей.

Первый опыт проводится следующим образом. Шприц (без иглы) заполняется теплой водой ($50\text{-}70^0$). Осуществляется прикосновением шприца к различным участкам кожи и фиксируется ощущение – тепло или холод.

Второй опыт. Кисть руки опускают в теплую воду. По секундомеру определяется (и фиксируется) время адаптации терморецепторов.

ОПРЕДЕЛЕНИЕ ПОРОГА ВКУСОВОЙ ЧУВСТВИТЕЛЬНОСТИ.

Рецепторы вкуса в основном расположены на сосочках языка. Некоторая часть вкусовых рецепторов локализуется в слизистой оболочке мягкого неба, миндалин, задней стенки глотки и надгорника, Различают четыре вида вкусовых рецепторов,- воспринимающих соленое, сладкое, горькое и кислое.

Определение порога вкусовой чувствительности осуществляется следующим образом. Первоначально приготовляют раствор соли различной концентрации. На язык

испытуемого капается раствор и фиксируется его концентрация в момент ощутимости. (Перед каждым экспериментом рот испытуемого ополаскивается водой).

9. ИССЛЕДОВАНИЕ ПОВЕДЕНИЯ ФУНКЦИЙ ПРИНАДЛЕЖНОСТЕЙ.

Цель работы: овладение навыками исследования поведение интегральной функции принадлежности, как критериальной составляющей нечетких систем управления, путем анализа темпа ее сходимости к определенной ассимптоде в зависимости от методики ее вычисления и ее составляющих.

Краткие теоретические сведения.

В теории автоматического управления существует достаточно много методов, позволяющих оптимизировать работу систем по тем или иным критериям качества при выполнении ряда ограничений.

Математический аппарат, используемый в традиционных методах автоматического управления, не всегда в полной мере может удовлетворить нуждам современного производства. Поэтому в последнее время находят широкое распространение так называемые “мягкие вычисления”, основной принцип которых заключается в обеспечении приемлемого (не обязательно оптимального) качества управления в условиях неопределённости при относительно невысоком уровне затрачиваемых ресурсов (стоимостных, временных, вычислительных и т. п.). К мягким вычислениям в настоящее время относят такие информационные технологии, как экспертные системы, нейронные сети, нечеткие системы, генетические алгоритмы и ряд других. В их основе лежит попытка некоторой формализации деятельности головного мозга человека и функционирования живых организмов.

Рассмотрим ряд аспектов использования нечеткой логики при управлении БТО (биотехническими объектами). Построение нечетких систем основано на имитации действия человека-оператора или ЛПР при помощи ЭВМ. Действительно, человеку свойственно оперировать не количественными показателями, а

качественными, но следует учитывать, что эти качественные понятия носят, по сути, нечёткий характер. При этом используются лингвистические переменные, описывающие входную ситуацию и управляющие воздействия на качественном уровне. Эти лингвистические переменные задаются на некоторой количественной шкале, при помощи которой определяются степени соответствия данных рассматриваемым понятиям. Для этого используются функции принадлежности, принимающие значения от 0 до 1. Возможные значения лингвистических переменных называются термами (например, для лингвистической переменной - «Уровень пульпы» термами являются «Понизить», «Повысить», «Не изменять»).

Кроме того, задается набор правил, ставящих в соответствие входной ситуации определённое управляющее воздействие. Эти правила обычно имеют вид «Если …, то…» и формируются при помощи эксперта или группы экспертов. Однако во многих случаях экспертам не удаётся принять однозначное решение о требуемом воздействии на объект при сложившейся ситуации. Например, если рассогласование между уставкой и выходом объекта отрицательно, а его производная положительна, даже опытному специалисту бывает затруднительно ответить, каким должно быть управляющее воздействие. Поэтому представляется целесообразным каждому правилу «Если …, то…» также ставить в соответствие некоторую величину из интервала от 0 до 1, отражающую степень уверенности в предпринимаемых действиях. Таким образом, формируется нечеткое соответствие между пространством предпосылок и пространством заключений.

Нечеткие системы управления оперируют с нечеткими множествами и нечеткими логическими выводами. На нечетких множествах определяются функции принадлежности. Интегральные функции принадлежности в качестве аргументов имеют частные функции принадлежности носителями которых являются различные характеристики объекта управления принимающие значения принадлежащих нечетким множествам.

Нечеткие множества – это частный случай нечисловых данных.

Пусть A - некоторое множество. Подмножество B множества A характеризуется своей характеристической функцией:

$$\mu_B(x) = \begin{cases} 1, & x \in B, \\ 0, & x \notin B. \end{cases} \qquad (1)$$

Что такое нечеткое множество? Обычно говорят, что нечеткое подмножество C множества A характеризуется своей функцией принадлежности $\mu_C : A \to [0;1]$. Значение функции принадлежности в точке x показывает степень принадлежности этой точки нечеткому множеству. Нечеткое множество описывает неопределенность, соответствующую точке x – она одновременно и входит, и не входит в нечеткое множество C. За вхождение - $\mu_C(x)$ шансов, за второе, т.е. за то, что точка не входит в множество, (1-$\mu_C(x)$) шансов.

Если функция принадлежности $\mu_C(x)$ имеет вид (1) при некотором B, то C есть обычное (четкое) подмножество A. Таким образом, теория нечетких множество является более общей или хотя бы не менее общей математической дисциплиной, чем обычная теория множеств, поскольку обычные множества – частный случай нечетких. Соответственно можно ожидать, что теория нечеткости как целое обобщает классическую математику.

По степени общности обычная математика и нечеткая математика эквивалентны. Однако для практического применения, например, в теории принятия решений описание и анализ неопределенностей с помощью теории нечетких множеств весьма плодотворны.

Теория нечеткости является обобщением интервальной математики. Действительно, функция принадлежности:

$$\mu_B(x) = \begin{cases} 1, & x \in [a;b], \\ 0, & x \notin [a;b] \end{cases}$$

задает интервальную неопределенность – про рассматриваемую величину известно лишь, что она лежит в заданном интервале [a,b].

Тем самым описание неопределенностей с помощью нечетких множеств является более общим, чем с помощью интервалов.

Л.А. Заде рассматривал теорию нечетких множеств как аппарат анализа и моделирования гуманистических систем, т.е. систем, в которых участвует человек. Его подход опирается на предпосылку о том, что элементами мышления человека являются не числа, а элементы некоторых нечетких множеств или классов объектов, для которых переход от «принадлежности» к «непринадлежности» не скачкообразен, а непрерывен. В настоящее время методы теории нечеткости используются почти во всех прикладных областях, в том числе при управлении предприятиями, качеством продукции и технологическими процессами, при описании предпочтений и оптимизации различных процессов, в том числе диагностического характера.

Л.А. Заде использовал термин «fuzzy set» (нечеткое множество). На русский язык термин «fuzzy» переводится как нечеткий, размытый, расплывчатый, и даже как пушистый и туманный.

В качестве примера аппарата теории нечетких множеств рассмотрим определения теоретико-множественных операций над нечеткими множествами. Пусть *C* и *D* - два нечетких подмножества *A* с функциями принадлежности $\mu_C(x)$ и $\mu_D(x)$ соответственно. Пересечением $C \cap D$, произведением *CD*, объединением $C \cup D$, отрицанием $\overline{C}$, суммой *C*+*D* называются нечеткие подмножества *A* с функциями принадлежности, соответственно:

$$\mu_{C\cap D}(x) = \min(\mu_C(x), \mu_D(x)), \quad \mu_{CD}(x) = \mu_C(x)\mu_D(x), \quad \mu_{\overline{C}}(x) = 1 - \mu_C(x),$$

$$\mu_{C\cup D}(x) = \max(\mu_C(x), \mu_D(x)), \quad \mu_{C+D}(x) = \mu_C(x) + \mu_D(x) - \mu_C(x)\mu_D(x), \quad x \in A,$$

Теория нечетких множеств в определенном смысле сводится к теории случайных множеств. Однако при решении прикладных задач вероятностно-статистические методы и методы теории нечеткости

Законы де Моргана для нечетких множеств. Законами же Моргана называются следующие тождества алгебры множеств:

$$\overline{A \bigcup B} = \overline{A} \bigcap \overline{B}, \quad \overline{A \bigcap B} = \overline{A} \bigcup \overline{B}. \qquad (2)$$

$$\overline{A \bigcup B} = \overline{A} \bigcap \overline{B}, \quad \overline{A \bigcap B} = \overline{A} \bigcup \overline{B}, \qquad (3)$$

$$\overline{A + B} = \overline{A}\,\overline{B}, \quad \overline{AB} = \overline{A} + \overline{B}. \qquad (4)$$

Тождества (3) и (4) называются *законами де Моргана для нечетких множеств*.

Дистрибутивный закон для нечетких множеств. Некоторые свойства операций над множествами не выполнены для нечетких множеств. Так, $A + A \neq A$, за исключением случая, когда A – «четкое» множество (т.е. функция принадлежности принимает только значения 0 и 1).

Носителем нечеткого множества A называется совокупность всех точек $y \in Y$, для которых $\mu_A(y) > 0$.

Нечетким логическим выводом (fuzzy logic inference) называется
получение заключения в виде нечеткого множества, соответствующего текущим значениям входов, с использованием нечеткой базы знаний и нечетких операций.

В общем случае нечеткий вывод решения происходит за три (или четыре) шага:

1) *этап фаззификации*. С помощью функций принадлежности всех термов входных лингвистических переменных и на основании задаваемых четких значений из универсумов входных лингвистических переменных определяются степени уверенности в том, что выходная лингвистическая переменная принимает конкретное значение.

2) *этап непосредственного нечеткого вывода*. На основании набора правил - нечеткой базы знаний - вычисляются значения истинности для предпосылок всех правил на основании конкретных нечетких операций, соответствующих конъюнкции или дизъюнкции термов в левой части правил. В большинстве случаев это либо максимум, либо минимум из степеней уверенности термов, вычисленных на этапе фаззификации, который применяется к заключению каждого правила. Используя один из способов построения нечеткой импликации, получается нечеткая переменная, соответствующая вычисленному значению степени уверенности в левой части правила и нечеткому множеству в правой части правила.

Основным способом построения нечеткой импликации является способ Kleene-Dienes: пусть A и B - нечеткие высказывания и $\mu_A \mu_B$ — соответствующие им функции принадлежности. Импликация A=>B определяется формулами:

$A \Rightarrow B \equiv \bar{A} \vee B$,

$\mu_{A \Rightarrow} B\ (x, y) = \max\ (1 - \mu_A\ (x);\ \mu_B\ (y))$.

Однако, это не единственное обобщение оператора импликации. В таблице приведены другие интерпретации этого понятия, наиболее часто применяемые на практике.

Larsen	$\mu_{A \Rightarrow B}\ (x, y) = \mu_A\ (x)\ \mu_B\ (y)$
Lukasiewicz	$\mu_{A \Rightarrow B}\ (x, y) = \min\ (1;\ 1 - \mu_A\ (x) + \mu_B\ (y))$
Mamdani	$\mu_{A \Rightarrow B}\ (x, y) = \max\ (\mu_A\ (x);\ \mu_B\ (y))$
Kleene-Dienes-Lu	$\mu_{A \Rightarrow B}\ (x, y) = 1 - \mu_A\ (x) + \mu_A\ (x)\ \mu_B\ (y)$

3) *этап композиции (агрегации, аккумуляции)*. Все нечеткие множества, назначенные для каждого терма каждой выходной лингвистической лингвистической переменной, обьединяются вместе, и формируется единственное нечеткое множество - значение для каждой выводимой лингвистической переменной. Обычно используются функции MAX или SUM.

4) *этап дефаззификации (необязательный)*. Используется тогда, когда полезно преобразовать нечеткий набор значений выводимых лингвистических переменных к точным. В теории нечетких множеств процедура дефузификации аналогична нахождению характеристик положения (математического ожидания, моды, медианы) случайных величин в теории вероятности. Имеется достаточно большое количество методов перехода к точным значениям.

Простейшим способом выполнения процедуры дефазификации является выбор четкого числа, соответствующего максимуму функции принадлежности. Однако пригодность этого способа распространяется лишь на одноэкстремальные функции принадлежности. Для многоэкстремальных функций принадлежности часто используются следующие методы дефузификации:

1) COG (Center Of Gravity) – «центр тяжести»". Физическим аналогом этой формулы является нахождение центра тяжести плоской фигуры, ограниченной осями координат и графиком функции принадлежности нечеткого множества.

2) MOM (Mean Of Maximums) – «центр максимумов». При использовании метода центра максимумов требуется найти среднее арифметическое элементов универсального множества, имеющих максимальные степени принадлежностей.

3) First Maximum – «первый максимум» - максимум функции принадлежности с наименьшей абсциссой.

Конкретные способы синтеза интегральных функций принадлежности определяется из характера решаемой задачи, ее целей и предметной области.

В связи с этим, на этапе разведочного анализа результатов мониторинга или в процессе диагностики целесообразно проводить исследование зависимости результирующей функции принадлежности от значений функции принадлежностей аргументов в нее входящих.

Порядок выполнения работы:

1. На абстрактном носителе x синтезировать функции принадлежности согласно Таблице 1.
2. Согласно Таблице 2 составить алгоритм вычисления интегральной функции принадлежности в случае применения N решающих правил обладающих определенными функциями принадлежностями.
3. Оценить значения скорости и ускорения сходимости интегральной функции к единице в зависимости от количества решающих правил и выбранной функции принадлежности.
4. Проанализировать полученные результаты с точки зрения: сходимости, применения определенных функций принадлежности и методов вычисления интегральной функции, количества итераций в процессе вычисления.
5. Сделать выводы.
6. Оформить отчет, включающий в себя результирующую таблицу вида:

	Способ получения интегрального показателя
Функция принадлежности	Скорость, ускорение, график

7. В процессе самостоятельной работы в ходе выполнения лабораторной работы рекомендуется проанализировать показатели вычисления интегральной функции принадлежности в случаях изменения в итерациях как правил вычисления функций принадлежности, так и правил вычисления интегральной функции.

Контрольные вопросы:

1. Чем отличается нечеткое множество от четкого?
2. Что такое функция принадлежности?
3. Как получить интегральное значение функции принадлежности в случае анализа нескольких нечетких множеств?
4. Что такое носитель функции принадлежности?
5. Когда рекомендуется применять аппарат теории нечетких множеств при построении диагностических правил?
6. В чем заключается алгоритмы нечеткого вывода Мамдани, Лоренса, Цукамото?

7. Что такое лингвистическая переменная?
8. Каким образом осуществляется коррекция интегральной функции принадлежности в случае неадекватно быстрого приближения ее значения к 1?

Библиография

1. Демидова, Лилия Анатольевна. Принятие решений в условиях неопределенности [Текст] : монография / Л. А. Демидова. - 2-е изд., перераб. - Москва : Горячая линия - Телеком, 2016. - 289 с.
2. Емельянов, Сергей Геннадьевич Интеллектуальные системы на основе нечеткой логики и мягких арифметических операций [Текст]: учебник / С. Г. Емельянов , В. С. Титов, М. В. Бобырь. - Москва: Аргамак-Медиа, 2014
3. Леоненков, А. Нечеткое моделирование в среде MATLAB и fuzzyTECH [Текст] /А.Леоненков.-СПб.:БХВ-Петербург,2003.-736с.
4. Нечеткие множества. Презентация / http://www.myshared.ru/slide/534083/
5. Рубанов, В. Г. Адаптивные системы принятия нечетко-логических решений [Текст]: монография / В. Г. Рубанов, В. С. Титов, М. В. Бобырь. - Белгород: БГТУ, 2014. - 236 с.
6. Формирование функций принадлежности нечеткой переменной . Презентация./http://knowledge.allbest.ru/management/3c0b65635b2bd78b5c53a89421216c26.html

Приложение

Таблица 1. – Типовые функции принадлежности

Название функции	Вид функции
S	$s(x;a,b,c)=\begin{cases} 0, для_x \le a \\ 2\cdot\left(\frac{x-a}{c-a}\right)^2, для_a \le x \le b \\ 1-2\cdot\left(\frac{x-c}{c-a}\right)^2, для_b \le x \le c \\ 1, для_x \ge c \end{cases}$
π	$\pi(x;b,c)=\begin{cases} s(x;c-b,c-b/2,c), _для_x \le c \\ 1-s(x;c,c+b/2,c+b), _для_x \ge c \end{cases}$

γ	$\gamma(x;a,b)=\begin{cases} 0,_для_x\le a \\ \dfrac{x-a}{b-a},_для_a\le x\le b \\ 1._для_x\ge b \end{cases}$
t	$t(x;a,b,c)=\begin{cases} 0,_для_x\le a \\ \dfrac{x-a}{b-a},_для_a\le x\le b \\ \dfrac{c-x}{c-b},_для_b\le x\le c \\ 0._для_\ge c \end{cases}$
L	$L(x;a,b)=\begin{cases} 1,_для_x\le a \\ \dfrac{b-x}{b-a},_для_a\le x\le b \\ 0,_x\ge b \end{cases}$

Таблица 2 Интегральные функции принадлежности

№	T(a,b) – «пересечение»	S(a,b) - «объединение»	параметры
1	Min(a,b)	Max(a,b)	
2	a*b	a+b-a*b	
3	MAX(a+b-1,0)	Min(a+b,1)	
4	$\begin{cases} a, & b=1 \\ b, & a=1 \\ 0, & èí\ à\div å \end{cases}$	$\begin{cases} a,_b=0 \\ b,_a=0 \\ 0,_a,b\ne 0 \end{cases}$	
5	$\dfrac{a\cdot b}{\gamma+(1-\gamma)\cdot(a+b-a\cdot b)}$	$\dfrac{a+b-(2-\gamma)\cdot a\cdot b}{1-(1-\gamma)\cdot(a\cdot b)}$	$\gamma>0$
6	$\dfrac{a\cdot b}{\max(a,b,\alpha)}$	$\dfrac{a+b-a\cdot b-\min(a,b,1-\alpha)}{\max(1-a,1-b,\alpha)}$	$\alpha\in[0,1]$
7	$1-\sqrt[p]{(1-a)^p+(1-b)^p-(1-a)^p\cdot(1-b)^p}$	$\sqrt[p]{a^p+b^p-a^p\cdot b^p}$	p>0
8	$[1+\sqrt[\lambda]{(\frac{1}{a}-1)^{\lambda}+(\frac{1}{b}-1)^{\lambda}}]^{-1}$	$[1+\sqrt[\lambda]{(\frac{1}{a}-1)^{-\lambda}+(\frac{1}{b}-1)^{-\lambda}}]^{-1}$	λ>0
9	$\max(1-\sqrt[p]{(1-a)^p+(1-b)^p},0)$	$\min(\sqrt[p]{a^p+b^p},1)$	$p\ge 1$
10	$\log_w(1+\dfrac{(w^a-1)\cdot(w^b-1)}{w-1})$	$1-\log_w(1+\dfrac{(w^{1-a})\cdot(w^{1-b})}{w-1})$	w= min(1,1-a; 1,1-b)
12	$\max(\dfrac{a+b-1+\lambda\cdot a\cdot b}{1+\lambda},0)$	$\min(a+b+\lambda\cdot a\cdot b,1)$	$\lambda\ge -1$

10. ПОСТРОЕНИЕ НЕЧЕТКОЙ СИСТЕМЫ С ИСПОЛЬЗОВАНИЕМ ПАКЕТА Fuzzy Logic Toolbox

Цель работы: приобретение навыков построения нечетких систем управления с помощью инструментария Fuzzy Logic Toolbox, реализованным в прикладном программном пакете Matlab.

Краткие теоретические сведения

В технических и биотехнических системах, применяемых в медицинской практике в силу специфики биообъекта нечеткая технология завоевывает все больше сторонников среди разработчиков систем управления поскольку позволяет осуществлять адекватное регулирование в условиях неопределенности. Термин «fuzzy» так вошел в жизнь, что на многих языках он даже не переводится.

Нечеткая логика основана на использовании таких оборотов естественного языка, как «далеко», «близко», «холодно», «горячо». Диапазон ее применения очень широк - от бытовых приборов до управления сложными промышленными процессами. Многие современные задачи управления просто не могут быть решены классическими методами из-за очень большой сложности математических моделей, их описывающих. Вместе с тем, чтобы использовать теорию нечеткости на цифровых компьютерах, необходимы математические преобразования, позволяющие перейти от лингвистических переменных к их числовым аналогам в ЭВМ. На рисунке 1 показаны области наиболее эффективного применения современных технологий управления. Как видно, классические методы управления хорошо работают при полностью детерминированном объекте управления и детерминированной среде, а для систем с неполной информацией и высокой сложностью объекта управления оптимальными являются нечеткие методы управления.

Классическая или булева логика имеет один существенный недостаток - с ее помощью невозможно описать ассоциативное

мышление человека. Классическая логика оперирует только двумя понятиями: ИСТИНА и ЛОЖЬ, исключая любые промежуточные значения. Аналогично этому булева логика не признает ничего кроме единиц и нулей. Все это хорошо для вычислительных машин, но недостаточно эффективно при проектировании и эксплуатации систем управления БТС.

Рис. 1. Области применения технологий управления

Решить эту проблему и призвана нечеткая логика. С термином «лингвистическая переменная» можно связать любую физическую величину, для которой нужно иметь больше значений, чем только ДА и НЕТ. В этом случае вы определяете необходимое число термов и каждому из них ставите в соответствие некоторое значение описываемой физической величины. Для этого значения степень принадлежности физической величины к терму будет равна единице, а для всех остальных значений - в зависимости от выбранной функции принадлежности.

Получившие наибольшее развитие из всех разработок искусственного интеллекта, экспертные системы завоевали устойчивое признание в качестве систем поддержки принятия

решений. Одним из основных методов представления знаний в экспертных системах являются продукционные правила, позволяющие приблизиться к стилю мышления человека. Любое правило продукций состоит из посылок и заключения. Главным же недостатком продукционных систем остается то, что для их функционирования требуется наличие полной информации о системе.

Нечеткие системы тоже основаны на правилах продукционного типа, однако в качестве посылки и заключения в правиле используются лингвистические переменные, что позволяет избежать ограничений, присущих классическим продукционным правилам.

Целевая установка процесса управления связывается с выходной переменной нечеткой системы управления, но результат нечеткого логического вывода является нечетким, а физическое исполнительное устройство не способно воспринять такую команду. Необходимы специальные математические методы, позволяющие переходить от нечетких значений величин к вполне определенным. В целом весь процесс нечеткого управления можно разбить на несколько шагов: фаззификация, разработка нечетких правил и дефаззификация.

Фаззификация (переход к нечеткости)

Точные значения входных переменных преобразуются в значения лингвистических переменных посредством применения некоторых положений теории нечетких множеств, а именно - при помощи определенных функций принадлежности.

Лингвистические переменные

В нечеткой логике значения любой величины представляются не числами, а словами естественного языка и называются ТЕРМАМИ. Так, значением лингвистической переменной ДИСТАНЦИЯ являются термы ДАЛЕКО, БЛИЗКО и т. д.

Функции принадлежности

Как уже говорилось, принадлежность каждого точного значения к одному из термов лингвистической переменной определяется посредством функции принадлежности. Ее вид может быть абсолютно произвольным. Стандартные функции принадлежности представлены на рисунке 2.

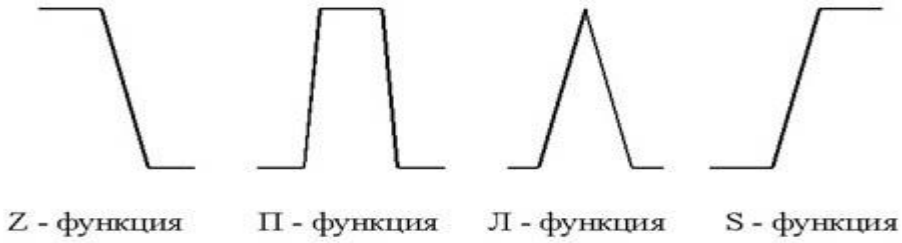


Рис. 2. – типовые функции принадлежности

Стандартные функции принадлежности легко применимы к решению большинства задач.

Разработка нечетких правил

На этом этапе определяются продукционные правила, связывающие лингвистические переменные. Совокупность таких правил описывает стратегию управления, применяемую в данной задаче. Большинство нечетких систем используют продукционные правила для описания зависимостей между лингвистическими переменными. Типичное продукционное правило состоит из антецедента (часть ЕСЛИ …) и консеквента (часть ТО …). Антецедент может содержать более одной посылки. В этом случае они объединяются посредством логических связок И или ИЛИ.
Процесс вычисления нечеткого правила называется нечетким логическим выводом и подразделяется на два этапа: обобщение и заключение.

Дефаззификация (устранение нечеткости)

На этом этапе осуществляется переход от нечетких значений величин к определенным физическим параметрам, которые могут

служить командами исполнительному устройству. Результат нечеткого вывода, конечно же, будет нечетким.

Для устранения нечеткости окончательного результата существует несколько методов. Рассмотрим некоторые из них.

Метод центра максимума (CoM)

Так как результатом нечеткого логического вывода может быть несколько термов выходной переменной, то правило дефаззификации должно определить, какой из термов выбрать.

Метод наибольшего значения (MoM)

При использовании этого метода правило дефаззификации выбирает максимальное из полученных значений выходной переменной.

Метод центроида (CoA)

В этом случае окончательное значение определяется как проекция центра тяжести фигуры, ограниченной функциями принадлежности выходной переменной с допустимыми значениями.

Основные шаги разработки нечеткой системы управления с использованием CAD-системы fuzzy TECH 3.0

Процесс разработки проекта нечеткой системы управления на fuzzy TECH разбивается, как уже говорилось, на четыре основных этапа. Все они схематично показаны на рисунке 3.

Рис. 3 – основные этапы разработки нечеткой системы управления.

Описание системы

На этом этапе при помощи средств, доступных в fuzzy TECH, задача формализуется. Здесь описываются: лингвистические переменные, их функции принадлежности, описать стратегия управления посредством нечетких правил. В целом CASE-технология, на основе которой построен пакет, позволяет все эти действия выполнить только посредством общения с экраном ЭВМ, не заглядывая в программный код.

Off-line-оптимизация

На этом этапе следует проверить работоспособность созданной системы посредством всех средств fuzzy TECH. Отметим, что можно использовать заранее созданный программный симулятор вашего объекта управления, подобно модели контейнерного крана. Для связи системы управления с моделью используется специально разработанный протокол связи fTlink, в основу которого положена концепция обмена сообщениями Windows. Все необходимые средства для установления связи с вашей моделью находятся в исходных текстах программ связи, поставляемых с пакетом.

On-line-оптимизация

На этом шаге разрабатываемая система управления и реальный объект управления соединяются физической линией связи (см. рисунок 4)

Рис. 4 – система управления при On-line-оптимизации

Такой вид отладки позволяет наблюдать поведение системы в реальных условиях и при необходимости вносить изменения в систему управления.

ФУНКЦИИ ПО РАБОТЕ С ПАКЕТОМ FUZZY LOGIC TOOLBOX

Редактор функций принадлежности.

(Membership Function Editor) редактор предназначен для задания следующей информации о терм-множествах входных и выходных переменных: количество термов; наименования термов; тип и параметры функций принадлежности, которые необходимы для представления лингвистических термов в виде нечетких множеств.

Редактор функций принадлежности может быть вызван из любого GUI-модуля (графического модуля среды Matlab), используемого с системами нечеткого логического вывода, командой **Membership Functions…** меню **Edit** или нажатием клавиш **Ctrl+2**.

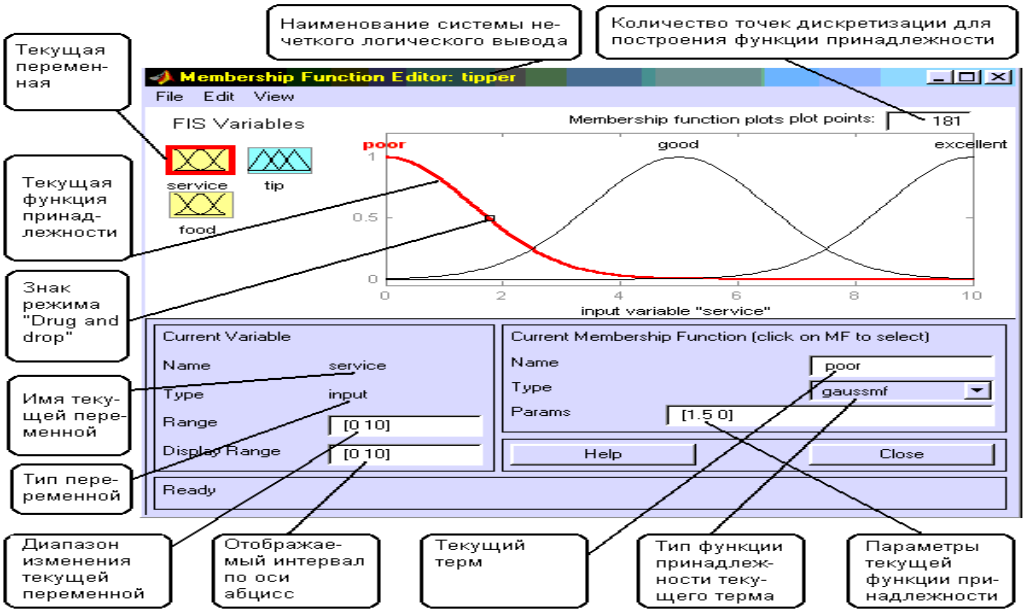


Рис. 5 – основное диалоговое окно

В FIS-редакторе открыть редактор функций принадлежности можно также двойным щелчком левой кнопкой мыши по полю входной или выходной переменных. Общий вид редактора функций принадлежности с указанием функционального назначения основных полей графического окна приведен на рис. В нижней части графического окна расположены кнопки **Help** и **Close**, которые позволяют вызвать окно справки и закрыть редактор, соответственно.

Редактор функций принадлежности содержит четыре меню - **File, Edit**, **View**, **Type** и четыре окна ввода информации – **Range, Display Range, Name** и **Params**. Эти четыре окно предназначены для задания диапазона изменения текущей переменной, диапазона вывода функций принадлежности, наименования текущего лингвистического терма и параметров его функции принадлежности, соответственно. Параметры функции принадлежности можно подбирать и в графическом режиме, путем изменения формы функции принадлежности с помощью технологии “Drug and drop”. Для этого необходимо

позиционировать курсор мыши на знаке режима “Drug and drop” (см. рис.), нажать на левую кнопку мыши и не отпуская ее изменять форму функции принадлежности. Параметры функции принадлежности будут пересчитываться автоматически.

Меню Edit.

Общий вид меню.

Рис. 6 -меню **Edit**

Команда **Undo** отменяет ранее совершенное действие. Выполняется также по нажатию **Ctrl+Z**. Команда **Add MFs…** позволяет добавить термы в терм-множество, используемое для лингвистической оценки текущей переменной. При выборе этой команды появится диалоговое окно, в котором необходимо выбрать тип функции принадлежности и количество термов. Значения параметров функций принадлежности будут установлены автоматически таким образом, чтобы равномерно покрыть область определения переменной, заданной в окне **Range**. При изменении области определения в окне **Range** параметры функций принадлежности будут промасштабированы.

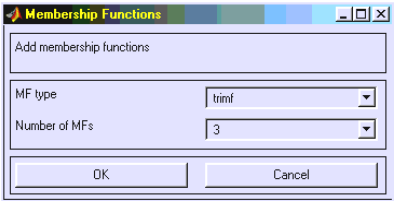


Рис. 7 - выбор количества термов и типа функций принадлежности.

Команда **Add Custom MF…** позволяет добавить один лингвистический терм, функция принадлежности которого

отличается от встроенных. После выбора этой команды появится графическое окно, в котором необходимо напечатать лингвистически терм (поле **MF name**),, имя функции принадлежности (поле **M-File function name**) и параметры функции принадлежности (поле **Parameter list**).

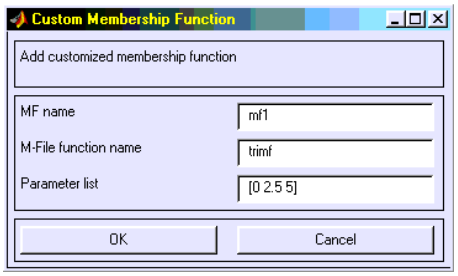


Рис. 8 - задание лингвистического терма с невстроенной функцией принадлежности.

Команда **Remove Selected MF** удаляет текущий терм из терм-множества текущей переменной. Признаком текущей переменной является красная окантовка ее прямоугольника. Признаком текущего терма является красный цвет его функции принадлежности. Для выбора текущего терма необходимо провести позиционирования курсора мыши на графике функции принадлежности и сделать щелчок левой кнопкой мыши.

Команда **Remove All MFs** удаляет все термы из терм-множества текущей переменной. Команда **FIS Properties…** открывает FIS-редактор. Эта команда может быть также выполнена нажатием **Ctrl+1**. Команда **Rules…** открывает редактор базы знаний. Эта команда может быть также выполнена нажатием **Ctrl+3**.

Меню Type

Это меню позволяет установить тип функций принадлежности термов, используемых для лингвистической оценки текущей переменной. На рис. приведено меню **Type**, в котором указаны возможные типы функций принадлежности.

Рис. 9 - Меню **Type**

Редактор базы знаний

Редактор базы знаний (**Rule Editor**) предназначен для формирования и модификации нечетких правил. Редактор базы знаний может быть вызван из любого GUI-модуля, используемого с системами нечеткого логического вывода, командой **Rules…** меню **Edit** или нажатием клавиш **Ctrl+3**. В FIS-редакторе открыть редактор базы знаний можно также двойным щелчком левой кнопкой мыши по прямоугольнику с названием системы нечеткого логического вывода, расположенного в центре графического окна.

Общий вид редактора базы знаний с указанием функционального назначения основных полей графического окна приведен на рис. В нижней части графического окна расположены кнопки **Help** и **Close**, которые позволяют вызвать окно справки и закрыть редактор, соответственно.

Редактор функций принадлежности содержит четыре системных меню **File, Edit**, **View**, **Options,** меню выбора термов входных и выходных переменных, поля установки логических операций И, ИЛИ, НЕ и весов правил, а также кнопки редактирования и просмотра правил.

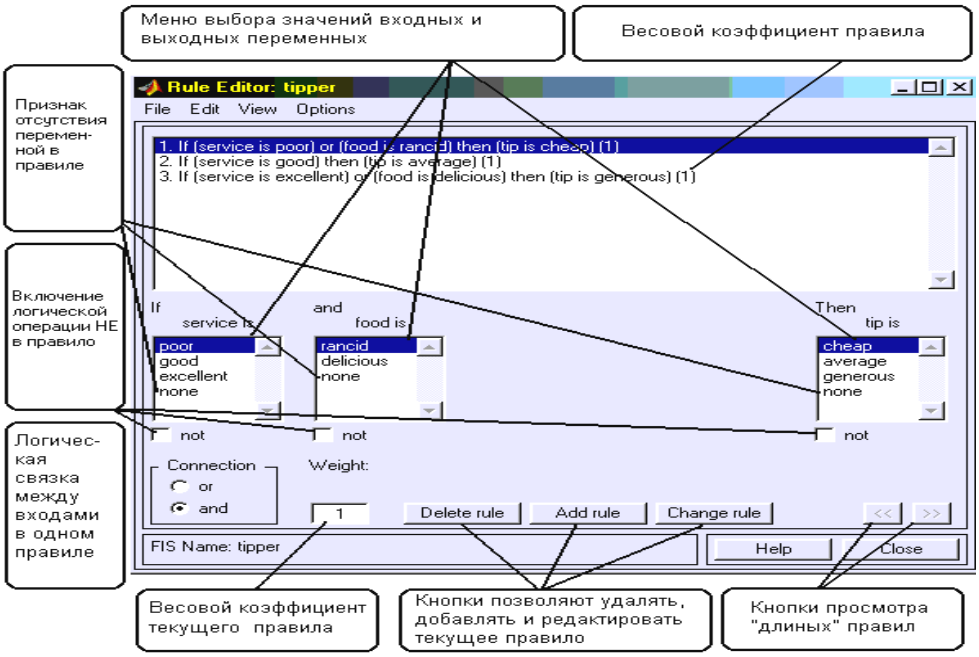


Рис. 10 – основное диалоговое окно редактора база знаний

Для ввода нового правила в базу знаний необходимо с помощью мыши выбрать соответствующую комбинацию лингвистических термов входных и выходных переменных, установить тип логической связки (И или ИЛИ) между переменными внутри правила, установить наличие или отсутствие логической операции НЕ для каждой лингвистической переменной, ввести значение весового коэффициента правила и нажать кнопку **Add Rule**. По умолчанию установлены следующие параметры: логическая связка переменных внутри правила – И; логическая операция НЕ – отсутствует; значение весового коэффициента правила – 1.

Возможны случаи, когда истинность правила не изменяется при произвольной значении некоторой входной переменной, т.е. это переменная не влияет на результат нечеткого логического вывода в данной области факторного пространства. Тогда в качестве лингвистического значения этой переменной необходимо установить **none.**

Для удаления правила из базы знаний необходимо сделать однократный щелчок левой кнопкой мыши по этому правилу и нажать кнопку **Delete Rule.**

Для модификации правила необходимо сделать однократный щелчок левой кнопкой мыши по этому правилу, затем установить необходимые параметры правила и нажать кнопку **Edit Rule.**

Меню Edit Общий вид меню приведен на рисунке 11.

Рисунок 11 - меню **Edit**

Команда **Undo** отменяет ранее совершенное действие. Выполняется также по нажатию **Ctrl+Z**. Команда **FIS Properties…** открывает FIS-редактор. Эта команда может быть также выполнена нажатием **Ctrl+1**. Команда **Membership Function…** открывает редактор функций принадлежностей. Эта команда может быть также выполнена нажатием **Ctrl+2**.

Меню Options

Это меню позволяет установить язык и формат правил базы знаний. При выборе команды **Language** появится список языков **English** (Английский), **Deutsch** (Немецкий), **Francais** (Французкий), из которого необходимо выбрать один. При выборе команды **Format** появится список возможных форматов правил базы знаний: **Verbose -** лингвистический; **Symbolic –** логический; **Indexed –** индексированный.

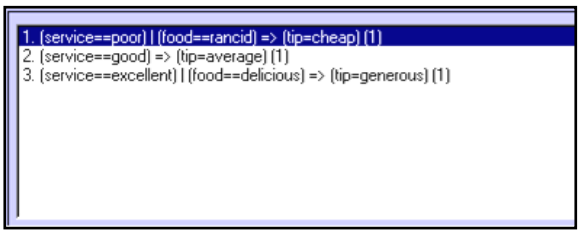


Рисунок 12 - база знаний в формате **Symbolic.**

Рисунок 13 - база знаний в формате **Indexed.**

Визуализация нечеткого логического вывода

Визуализация нечеткого логического вывода осуществляется с помощью GUI-модуля **Rule Viewer**. Этот модуль позволяет проиллюстрировать ход логического вывода по каждому правилу, получение результирующего нечеткого множества и выполнение процедуры дефаззификации. **Rule Viewer** может быть вызван из любого GUI-модуля, используемого с системами нечеткого логического вывода, командой **View rules …** меню **View** или нажатием клавиш **Ctrl+4**. Вид **Rule Viewer** с указанием функционального назначения основных полей графического окна приведен на рисунке 14.

Rule Viewer содержит четыре меню - **File, Edit**, **View**, **Options,** два поля ввода ввода информации – **Input** и **Plot points** и кнопки прокрутки изображения влево-вправо (**left-right)**, вверх-вниз (**up-down**). В нижней части графического окна расположены также кнопки **Help** и **Close**, которые позволяют вызвать окно справки и закрыть редактор, соответственно.

Каждое правило базы знаний представляется в виде последовательности горизонтально расположенных прямоугольников. При этом первые два прямоугольника отображают функции принадлежностей термов посылки правила (ЕСЛИ-часть правила), а последний третий прямоугольник соответствует функции принадлежности терма-следствия выходной переменной (ТО-часть правила). Пустой прямоугольник в

визуализации второго правила означает, что в этом правиле посылка по переменной food отсутствует (food is none). Желтая заливка графиков функций принадлежностей входных переменных указывает насколько значения входов, соответствуют термам данного правила. Для вывода правила в формате **Rule Editor** необходимо сделать однократный щелчок левой кнопки мыши по номеру соответствующего правила. В этом случае указанное правило будет выведено в нижней части графического окна.

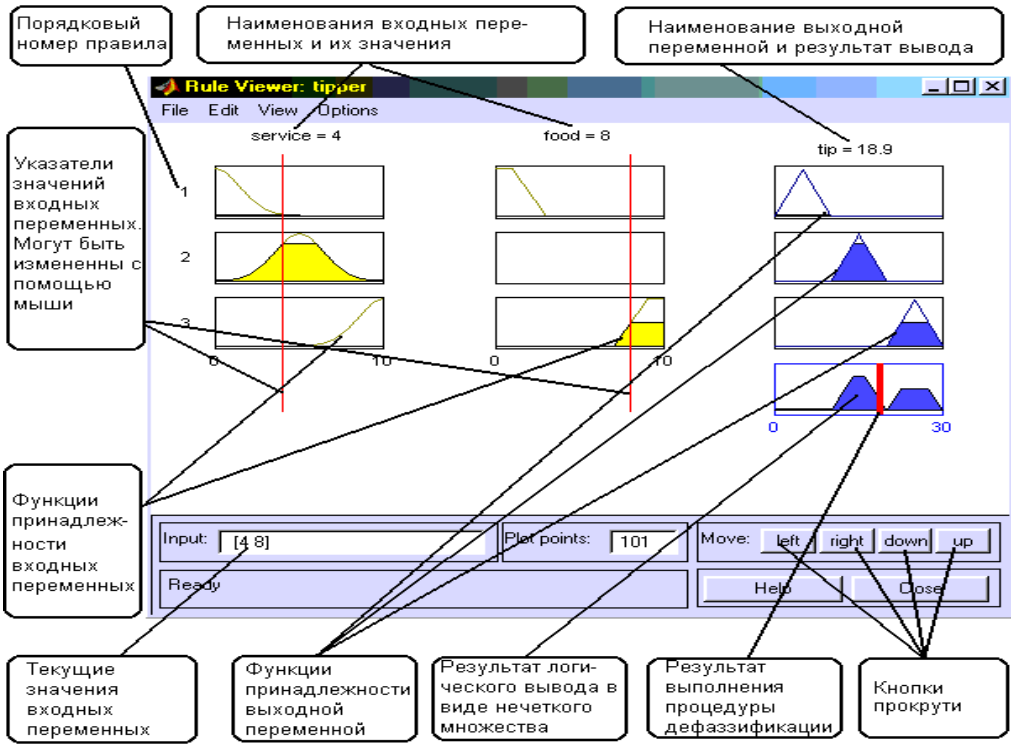


Рис. 14 – основное меню визуализации логического вывода с помощью **Rule Viewer**

Голубая заливка графика функции принадлежности выходной переменной представляет собой результат логического вывода в виде нечеткого множества по данному правилу. Результирующее нечеткое множество, соответствующее логическому выводу по всем правилам показано в нижнем прямоугольнике последнего

столбца графического окна. В этом же прямоугольнике красная вертикальная линия соответствует четкому значению логического вывода, полученного в результате дефаззификации.

Ввод значений входных переменных может осуществляться двумя способами:

- путем ввода численных значений в поле **Input**;
- с помощью мыши, путем перемещения линий-указателей красного цвета.

В последнем случае необходимо позиционировать курсор мыши на красной вертикальной линии, нажать на левую кнопку мыши и не отпуская ее переместить указатель на нужную позицию. Новое численное значения соответствующей входной переменной будет пересчитано автоматически и выведено в окно **Input**.

В поле **Plot points** задается количество точек дискретизации для построения графиков функций принадлежности.

Меню Edit Общий вид меню приведен на рисунке 15.

Рис. 15 – общий вид меню.

Команда **FIS Properties…** открывает FIS-редактор. Эта команда может быть также выполнена нажатием **Ctrl+1**. Команда **Membership Functions…** открывает редактор функций принадлежностей. Эта команда может быть также выполнена нажатием **Ctrl+2**. Команда **Rules…** открывает редактор базы знаний. Эта команда может быть также выполнена нажатием **Ctrl+3**.

Меню Options

Меню **Options** содержит только одну команду **Format**, которая позволяет установить один из следующих форматов вывода

выбранного правило в нижней части графического окна: **Verbose** - лингвистический; **Symbolic** – логический; **Indexed** – индексированный.

Визуализация поверхности «входы-выход»

Визуализация поверхности “входы-выход” осуществляется с помощью GUI-модуля **Surface Viewer**. Этот модуль позволяет вывести графическое изображение зависимости значения любой выходной переменной от произвольных двух (или одной) входных переменных. **Surface Viewer** может быть вызван из любого GUI-модуля, используемого с системами нечеткого логического вывода, командой **View surface …** меню **View** или нажатием клавиш **Ctrl+4**. Общий вид модуля **Surface Viewer** с указанием функционального назначения основных полей графического окна приведен на рисунке 16.

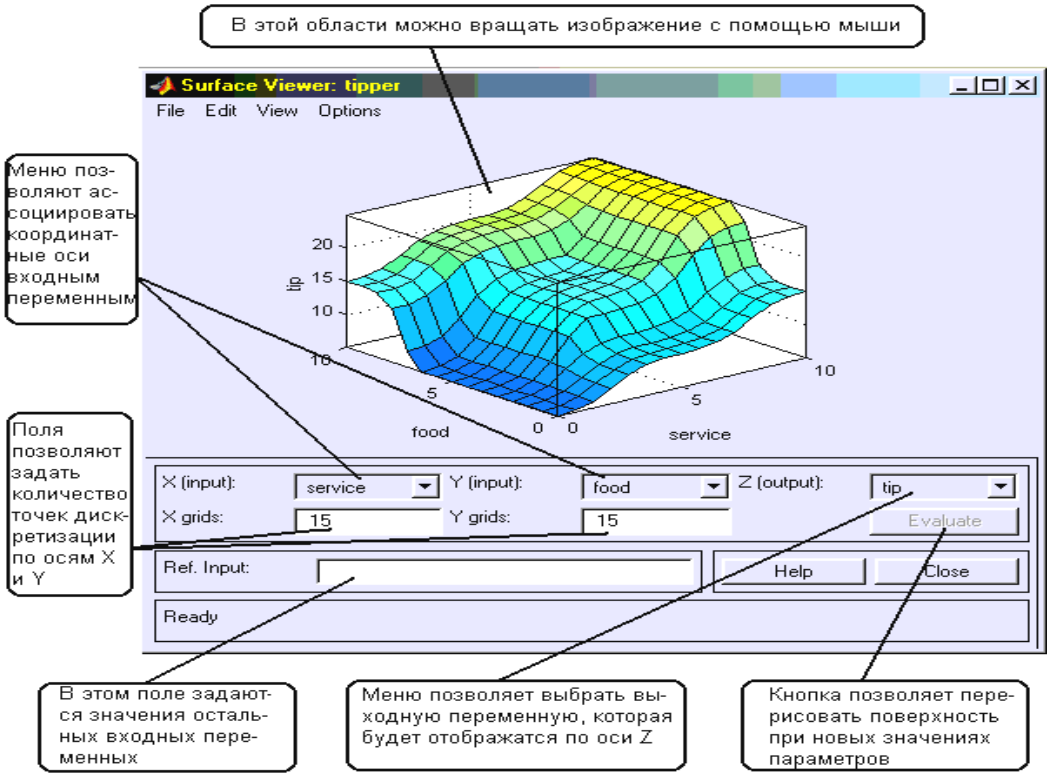
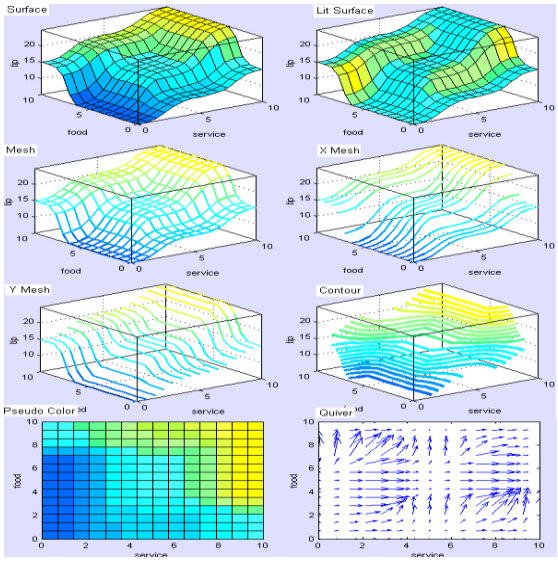


Рис. 16 – основное меню визуализация поверхности «входы-выход» с помощью Surface Viewer.

Surface Viewer содержит верхних системных меню - **File, Edit**, **View**, **Options,** три меню выбора координатных осей - **X (input)**, **Y (input)**, **Z (output)**, три поля ввода ввода информации – **X girds**, **Y girds**, **Ref. Input** и кнопку **Evaluate** для построения поверхности при новых параметрах. В нижней части графического



окна расположены также кнопки **Help** и **Close**, которые позволяют вызвать окно справки и закрыть редактор, соответственно.

Surface Viewer позволяет вращать поверхность «входы-выход» с помощью мыши. Для этого необходимо позиционировать курсор мыши на поверхности «входы-выход», нажать на левую кнопку мыши и не отпуская ее повернуть графическое изображение на требуемый угол.

Поля **X girds** и **Y girds** предназначены для задания количества точек дискретизации по осям **X** и **Y**, для построения поверхности «входы-выход». По умолчанию количество дискрет по каждой оси равно 15. Для изменения этого значения необходимо установить маркер на поле **X girds** (**Y girds**) и ввести новое значение.

Поле **Ref. Input** предназначено для задания значений входных переменных, кроме тех, которые ассоциированы с координатными осями. По умолчанию это значения середины интервалов изменения переменных. Для изменения этого значения необходимо установить маркер на поле **Ref. Input** и ввести новые значение.

Меню координатных осей

Меню **X (input)**, **Y (input)**, **Z (output)** позволяют поставить в соответствие осям координат входные и выходные переменные.

При этом входные переменные могут отображаться только по осям **X** и **Y**, а выходные переменные только по оси **Z**. В **Surface Viewer** предусмотрена возможность построения однофакторных зависимостей “вход-выход”. Для этого в меню второй координатной оси (**X (input)** или **Y (input)**) необходимо выбрать **none**.

Меню Options Меню **Options** изображено на рисунке 17. Оно содержит команды **Plot**, **Color Map** и **Always evaluate.**

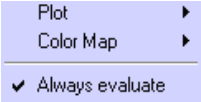


Рис. 17 – общий вид меню.

Команда **Plot** позволяет управлять форматом вывода поверхности «входы-выход». При выборе этой команды появляется меню (рисунок 18), в котором необходимо выбрать формат вывода поверхности.

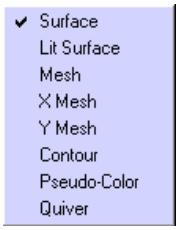


Рис. 18 – меню команды Plot.

Команда **Color Map** позволяет управлять палитрой цветов при выводе поверхности «входы-выход». При выборе этой команды появляется меню, в котором необходимо выбрать одну из палитр: ***default*** *– использовать палитру, установленную по умолчанию;* ***blue*** *– холодная сине-голубая палитра;* ***hot*** *– теплая палитра, состоящая из черного, красного, желтого и белого цветов;* ***HSV*** *– палитра насыщенных цветов: красный, желтый, зеленый, циан, голубой, мажента, красный.*

Команда **Always evaluate** позволяет установить или отменить режим автоматического, т.е. без нажатия кнопки **Evaluate**,

перерисовывания поверхности «входы-выход» при любом изменении параметров.

Демонстрационный пример

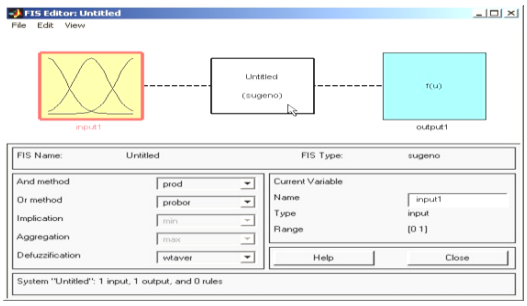
Требуется сконструировать нечеткую систему, отображающую зависимость между двумя переменными x и y. Зависимость y = f(x) задана пятью точками, приведенными в таблице.

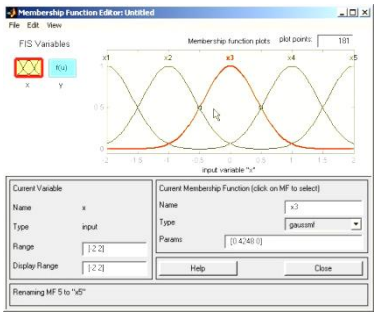
	-2	-1	0	1	2
y	-8	-1	0	1	8

Опишем действия необходимые для построения интерполяции данной функции. Для решения задания будем использовать пакет Fuzzy Logic Toolbox (пакет нечеткой логики), входящий в состав MatLab. Этот пакет позволяет конструировать нечеткие экспертные или управляющие системы.

Откроем пункт меню Start/ Toolboxes/Fuzzy Logic/FIS Editor.

Создаем новую нечеткую систему вывода типа Sugeno: File/ New FIS /Sugeno.



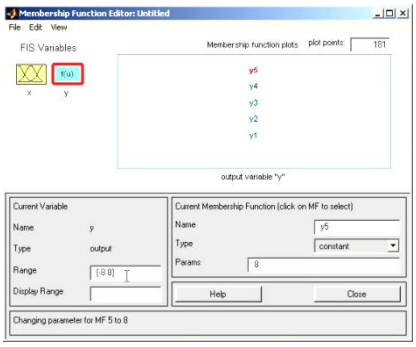


Далее в редакторе функций принадлежности добавим пять новых гауссовых функций принадлежности (для входного сигнала) при помощи Edit/ Add variables…/Input .

В поле Диапазон (Range) задаем диапазон изменения переменной x от -2 до 2, соответственно таблице.

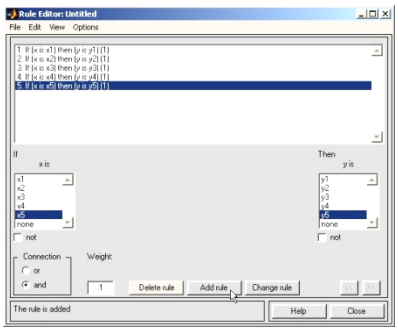
Для успешного решения поставленной задачи необходимо, чтобы ординаты максимумов функций принадлежности совпадали с заданными значениями аргумента x. Смещая графики функций принадлежности по горизонтали, добиваемся выполнения этого условия.

Зададим теперь функции принадлежности для выходной переменной y. Добавим 5 постоянных функций принадлежности (5 – число различных значений выходной переменной y, постоянные функции соответствуют алгоритму Sugeno).

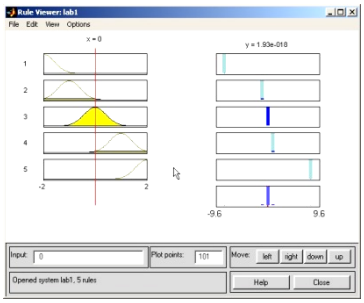


Зададим диапазон изменения выходной переменной от -8 до 8 и для каждой переменной зададим значения из таблицы.

В редакторе правил необходимо сконструировать соответствующие правила. При вводе каждого правила устанавливаем соответствие между функцией принадлежности аргумента x и числовым значением y.



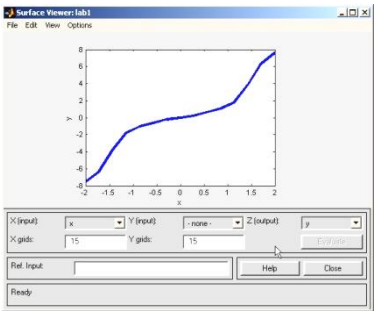
Выберем в пунктах меню Просмотр правил и изучим добавленные зависимости:



В правой части окна в графической форме представлены функции принадлежности аргумента x, а в левой – переменной выхода y с пояснением механизма принятия решений.

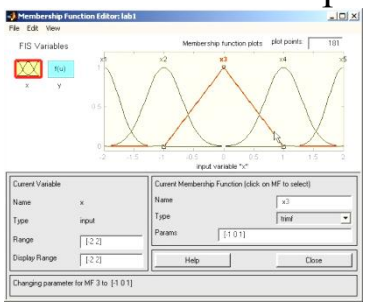
Перемещая красную черту или изменяя значения аргумента x в правой части окна, получаем значения переменной y в левой части окна. Таким образом, получено решение задачи интерполяции функции, заданной координатами пяти точек.

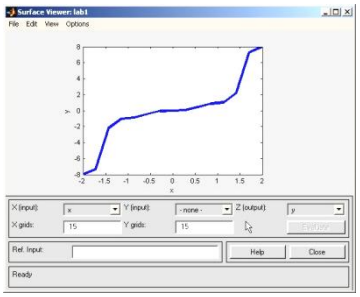
В окне просмотра поверхности отклика (выхода) можно визуально изучить полученный график функции y = f(x).



Как видно из приведенного графика функция получилась недостаточно четкой. Это обусловлено небольшим количеством исходных точек.

Но изменяя параметры функций принадлежности можно добиться более оптимального результата.





Порядок выполнения работы

1. Выпишите задание согласно варианта из таблицы «Индивидуальные задания». Рассчитайте значения двенадцати последовательных значений для заданной функции.

2. Составте нечеткую экспертную систему, которая бы аппроксимировала функцию по таблицам значений функции и аргумента, составленной в пункте 1.
3. Смоделируйте, как изменится выходное значение при изменении выходов.
4. Рассчитайте ответ нечеткой системы «вручную» и проверьте его при помощи созданной НЭС.
5. Оформите отчет.

Индивидуальные задания.

№ варианта	Функция f(x)
1	$\frac{x^2}{2}+5\cos(x)$
2	$2e^x-\frac{5}{2}x^2$
4	$\frac{x^2}{(x-1)^2}+x^4$
5	$\sqrt{x+x^2}+\frac{1}{1+x^2}$
6	$arctg(\sin(x)-\cos(x))$
7	$\frac{x^2-6x+4}{3x-2}+e^{2x}$
9	$sh^2(x)+e^x$
10	$\frac{1}{\ln(x)}+x^2$
12	$\frac{1}{x^2}+tg(x)$
13	$\sin(x)^2+\sin^2(x)$
14	$\frac{1}{sh(x)}+x^4$
15	$\frac{e^{x-3}}{x-3}+\ln(x)$

Контрольные вопросы:

1. Когда рекомендуется применять нечеткие системы управления?
2. Какие наиболее часто употребляемые функции принадлежности?
3. Как реализуется механизм фузификации?
4. Как реализуется механизм дефузикации?
5. Каким образом задаются тип и параметры функций принадлежности в среде MATLAB?
6. В чем заключается метод центра максимума ?
7. В чем заключается метод наибольшего значения ?
8. В чем заключается метод центроида?
9. Каким образом осуществляется переход от нечеткого множества к четному?
10. В чем заключаются основные этапы синтеза нечеткой системы управления?

Библиография.

1. Березин С.Я. Основы кибернетики и управление в биологических и медицинских системах: учебное пособие / С.Я. Березин. – Старый Оскол: ТНТ, 2013. – 244 с.
2. Борисов В.В., Круглов В.В., Федулов А.С. Нечеткие модели и сети. М.: Горячая линия – Телеком, 2017.-284 с.
3. Дьяконов В.П., Круглов В.В. MATLAB 6/5 SP1/7/7 SP 5/6. Инструменты искусственного интеллекта и биоинформатика. Серия «Библиотека профессионала».-М.: СОЛОН-ПРЕСС, 2006. -456 с.
4. Емельянов, С. Г. Автоматизированные нечетко-логические системы управления [Текст]: монография / С. Г. Емельянов, В. С. Титов, М. В. Бобырь. - М.: ИНФРА-М, 2011. - 176 с. - (Научная мысль).
5. Методы робастного, нейро-нечеткого и адаптивного управления [Текст]: учебник для студентов вузов / ред. Н. Д. Егупов. - М. : МГТУ, 2001. - 743 с.

11. ИСПОЛЬЗОВАНИЕ НЕЙРОННОЙ СЕТИ ПРИ РЕШЕНИИ КЛАССИФИКАЦИОННЫХ ЗАДАЧ СОСТОЯНИЯ ОБЪЕКТА УПРАВЛЕНИЯ.

Цель работы: овладение навыками применения искусственных нейронных сетей в диагностическом процессе на примере персептрона и сетей Кохонена.

Краткие теоретические сведения.

Нейронные сети представляют собой новую и весьма перспективную вычислительную технологию, дающую новые подходы к исследованию динамических и классификационных задач в системах поддержки принятия решений в диагностическом процессе и-или при обработке результатов мониторинга функционирования биообъекта (включая экологические аспекты).

Способность к моделированию нелинейных процессов, работе с зашумленными данными и адаптивность дают возможности применять нейронные сети для решения широкого класса задач.

Приложения нейронные сетей охватывают самые разнообразные области интересов: распознавание образов, обработка зашумленные данных, дополнение образов, ассоциативный поиск, классификация, оптимизация, прогноз, диагностика, обработка сигналов, абстрагирование, управление процессами, сегментация данных, сжатие информации, сложные отображения, моделирование сложных процессов, машинное зрение, распознавание речи.

Первый интерес к нейросетям был обусловлен пионерской работой МакКаллока и Питса, изданной в 1943 году, где предлагалась схема компьютера, основанного на аналогии с работой человеческого мозга. Они создали упрощенную модель нервной клетки – *нейрон.* Мозг человека состоит из белого и серого веществ: белое – это тела нейронов, а серое – это соединительная ткань между нейронами, или *аксоны и дендриты.* Мозг состоит примерно из 10^{11} нейронов, связанных между собой. Каждый нейрон получает информацию через свои дендриты, а передает ее дальше только через единственных аксон,

разветвляющийся на конце на тысячи *синапсов.* Простейший нейрон может иметь до 10000 дендритов, принимающих сигналы от других клеток. Таким образом, мозг содержит не менее 10^{15} взаимосвязей между нейронами, обеспечивающих его целевое востребуемое функционирование.

Искусственным нейроном называется простой элемент, сначала вычисляющий взвешенную сумму V входных величин x_i :

$V = \sum_{i=1}^{N} W_i * x_i = \overline{W} * \overline{X}$, где N – размерность пространства входных сигналов.

Затем полученная сумма сравнивается с пороговой величиной W_0, вслед за чем вступает в действие нелинейная функция активации f. Коэффициенты $\{W_i\}$ во взвешенной сумме обычно называют *синаптическими коэффициентами* или *весами*. Саму же взвешенную сумму V мы будем называть потенциалом нейрона *i*. Выходной сигнал тогда имеет вид f(V).

Величину порогового барьера можно рассматривать как еще один весовой коэффициент при постоянном входном сигнале. В этом случае говорится о *расширенном входном пространстве*: нейрон с N-мерным входом имеет N+1 весовой коэффициент. Если ввести в уравнение пороговую величину W_0, то оно перепишется так:

$$V = \sum_{i=1}^{N} W * x + Wo.$$

В зависимости от способа преобразования сигнала и характера активации возникают различные виды нейронных структур. Существуют *детерминированные нейроны,* когда активизирующая функция однозначно вычисляет выход по входу, и *вероятностные нейроны,* состояние которых в момент t есть случайная функция потенциала и состояния в момент t-1.

В искусственных нейронах могут быть различные функции активации, к наиболее типовым из которых относятся следующие виды функций:

♦ Линейная: выходной сигнал нейрона равен его потенциалу,

♦ пороговая: нейрон выбирает решение из двух вариантов: активен / неактивен,

♦ Многопороговая: выходной сигнал может принимать одно из q значений, определяемых (q-1) порогом внутри предельных значений.

♦ Сигмоидная: рассматриваются два вида сигмоидных функций:

$$s = f(V) = \frac{1}{1+\exp(-bV)}$$

с выходными значениями в промежутке [0,1] и

$$s = f(V) = \frac{\exp(bV)-1}{\exp(bV)+1}$$

с выходными значениями в промежутке [-1,1].

Коэффициент b определяет *крутизну* сигмоида. Поскольку сигмоидная функция является гладким отображением (-∞,∞) на (-1,1), то крутизну можно учесть через величины весов и порогов, и без ограничения общности можно полагать ее равной единице.

Графические изображения простейшего нейрона и виды функций с их графиками приведены на рисунке 1.

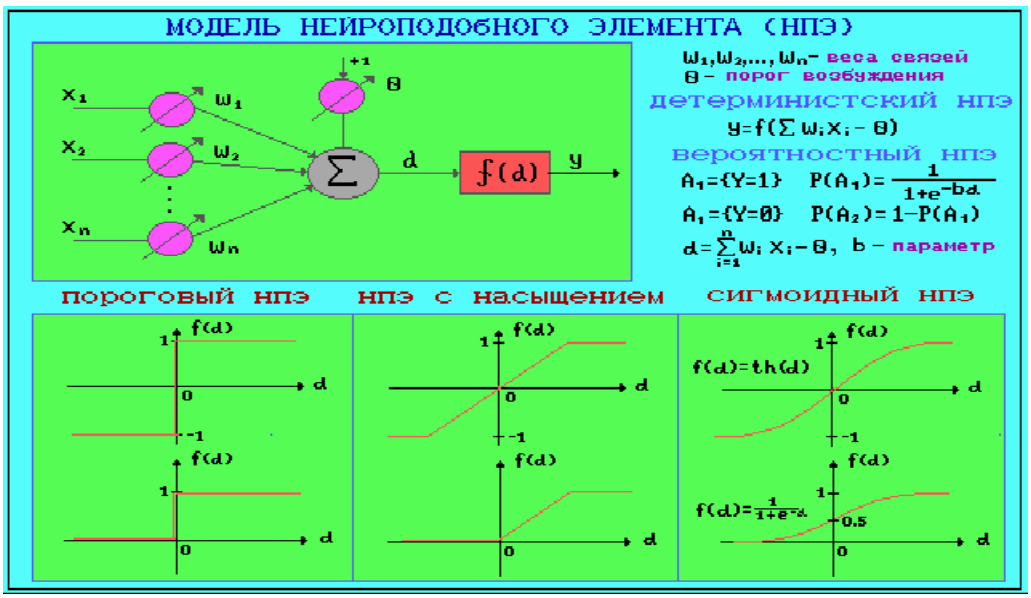


Рис.1 Изображение искусственного нейрона

Из точек на плоскости и соединений между ними можно построить множество графических фигур, называемых *графами*. Если каждую точку представить себе как один нейрон, а соединения между точками – как дендриты и синапсы, то получим нейронную сеть. Но не всякое соединение нейронов будет работоспособно или вообще целесообразно. По архитектуре связей нейросети могут быть сгруппированы в два класса: сети прямого распространения, в которых связи не имеют петель, и сети рекуррентного типа, в которых возможны обратные связи(см. рисунок 2).

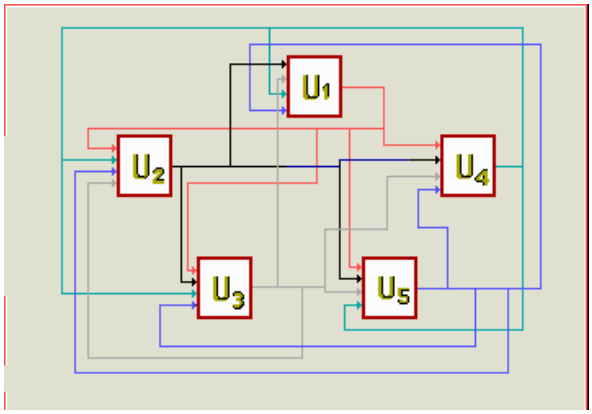


Рис. 2 Рекуррентная сеть

Сети прямого распространения подразделяются на однослойные перцепротроны (сети) и многослойные перцептроны (сети). Название перцептрона для нейросетей придумал американский нейрофизиолог Ф. Розенблатт , придумавший в 1957 году первый нейропроцессорный элемент (НПЭ) , то есть нейросеть. Он же доказал сходимость области решений для перцептрона при его обучении. Сразу после этого началось бурное исследование в этой области и был создан самый первый нейрокомпьютер Mark I. Многослойные сети отличаются тем, что между входными и выходными данными располагаются несколько так называемых скрытых слоев нейронов, добавляющих больше нелинейных связей в модель.

Рассмотрим устройство простейшей многослойной нейросети (схема представлена на рисунке 3). Любая нейронная

сеть состоит из **входного слоя** и **выходного слоя.** Соответственно подаются независимые и зависимые переменные. Входные данные преобразуются нейронами сети и сравниваются с выходом. Если отклонение больше заданного, то специальным образом изменяются веса связей нейронов между собой и пороговые значения нейронов. Снова происходит процесс вычислений выходного значения и его сравнение с эталоном. Если отклонения меньше заданной погрешности, то процесс обучения прекращается.

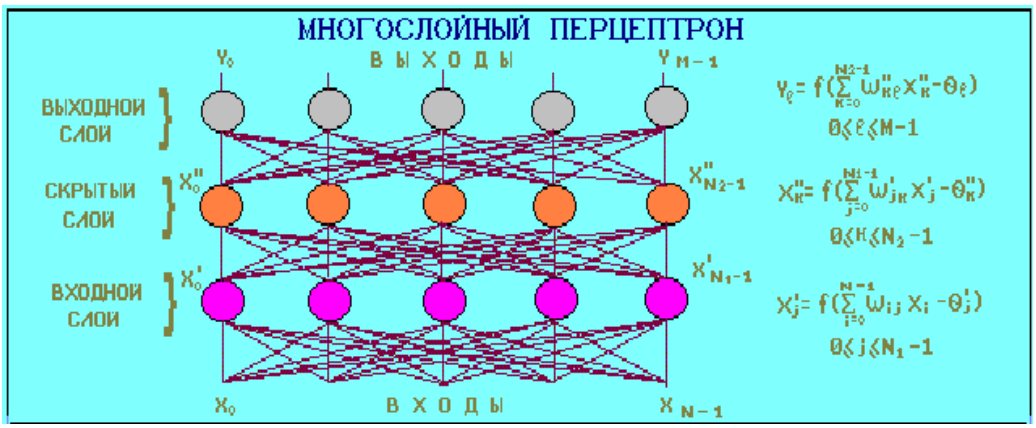


Рис. 3 Схема многослойного нейрона

Помимо входного и выходного слоев в многослойной сети существуют так называемые **скрытые слои.** Они представляют собой нейроны, которые не имеют непосредственных входов исходных данных, а связаны только с выходами входного слоя и с входом выходного слоя. Таким образом, скрытые слои дополнительно преобразуют информацию и добавляют нелинейности в модели.

Рассмотрим более подробно функционирование многослойного персептрона, представленного на рис. 3.

Если однослойная нейросеть очень хорошо справляется с задачами классификации, так как выходной слой нейронов сравнивает полученные от предыдущего слоя значения с порогом и выдает значение либо ноль, то есть меньше порогового значения, либо единицу - больше порогового (для случая пороговой внутренней функции нейрона), и не способен решать

большинство практических задач (что было доказано Минским и Пейпертом), то многослойный перцептрон с сигмоидными решающими функциями **способен аппроксимировать любую функциональную зависимость** (это было доказано в виде теоремы). Но при этом не известно ни нужное число слоев, ни нужное количество скрытых нейронов, ни необходимое для обучения сети время. Эти проблемы до сих пор стоят перед исследователями и разработчиками нейросетей. Поведение рекуррентных сетей описывается дифференциальными или разностными уравнениями, как правило, первого порядка. Это расширяет области применения нейросетей и способы их обучения. Сеть организована так, что каждый нейрон получает входную информацию от других нейронов, возможно, и от самого себя, и от окружающей среды. Этот тип сетей имеет важное значение, так как с их помощью можно моделировать нелинейные динамические системы.

Среди рекуррентных сетей можно выделить **сети Хопфилда и сети Кохонена.**

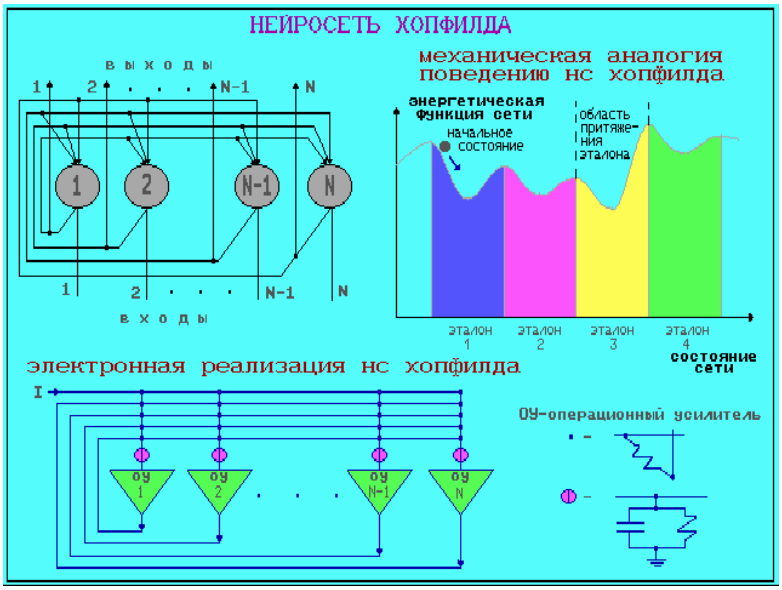


Рис.4 Структура нейросети Хопфилда

С помощью сетей Хопфилда можно обрабатывать неупорядоченные (рукописные буквы), упорядоченные во времени

(временные ряды) или пространстве (графики) образцы. Рекуррентная нейросеть простейшего вида была введена Хопфилдом и построена она из N нейронов, связанных каждый с каждым кроме самого себя, причем все нейроны являются выходными.

Нейросеть Хопфилда можно использовать в качестве **ассоциативной памяти.** Архитектура сети Хопфилда изображена на рис. 4. Сеть Кохонена еще называют «самоорганизующейся картой признаков». Сеть такого типа рассчитана на самостоятельное обучение во время обучения сообщать ей правильные ответы необязательно. В процессе обучения на вход сети подаются различные образцы. Сеть улавливает особенности их структуры и разделяет образцы на кластеры, а уже обученная сеть относит каждый вновь поступающий пример к одному из кластеров, руководствуясь некоторым критерием «близости». Сеть состоит из одного входного и одного выходного слоя. Количество элементов в выходном слое непосредственно определяет, сколько различных кластеров сеть сможет распознать. Каждый из выходных элементов получает на вход весь входной вектор. Как и во всякой нейронной сети, каждой связи приписан некоторый синаптический вес. В большинстве случаев каждый выходной элемент соединен также со своими соседями. Эти внутрислойные связи играют важную роль в процессе обучения, так как корректировка весов происходит только в окрестности того элемента, который наилучшим образом откликается на очередной вход. Выходные элементы соревнуются между собой за право вступить в действие и «получить урок». Выигрывает тот из них, чей вектор весов окажется ближе всех к входному вектору.

Главное отличие и преимущество нейросетей перед классическими средствами прогнозирования и классификации заключается в их способности к обучению. На этапе обучения происходит вычисление синаптических коэффициентов в процессе решения нейронной сетью задач, в которых нужный ответ определяется не по правилам, а с помощью примеров, сгруппированных в обучающие множества. Таким образом, нейросеть на этапе обучения сама выполняет роль *эксперта* в

процессе подготовки данных для построения экспертной системы. Предполагается, что *правила* находятся в структуре обучающих данных.

Для обучения нейронной сети требуются *обучающие данные*. Они должны отвечать свойствам ***представительности и случайности или последовательности***. Все зависит от класса решаемой задачи. Такие данные представляют собой ряды примеров с указанием для каждого из них значением выходного параметра, которое было бы желательно получить. Действия, которые при этом происходят, можно назвать *контролируемым обучением*: «учитель» подает на вход сети вектор исходных данных, а на выходной узел сообщает желаемое значение результата вычислений. Контролируемое обучение нейросети можно рассматривать как решение оптимизационной задачи. Ее целью является минимизация функции ошибок E на данном множестве примеров путем выбора значений весов W. Достижение минимума называется *сходимостью* процесса обучения. Именно возможность этого и доказал Розенблатт. Поскольку ошибка зависит от весов нелинейно, получить решение в аналитической форме невозможно, и поиск глобального минимума осуществляется посредством итерационного процесса - так называемого *обучающего алгоритма*. Разработано уже более сотни разных обучающих алгоритмов, отличающихся друг от друга стратегией оптимизации и критерием ошибок. Обычно в качестве меры погрешности берется средняя квадратичная ошибка (СКО):

$E = \sqrt{\frac{\sum_{i=1}^{M}(d_i - y_i)^2}{M}}$, где M – число примеров в обучающем множестве.

Минимизация величины E осуществляется с помощью градиентных методов. Изменение весов происходит в направлении, обратном к направлению наибольшей крутизны для функции:

$W(t+1) = W(t) + \varepsilon * \frac{\partial E}{\partial W}$, где ε - определяемый пользователем параметр, который называется коэффициентом обучения.

Одним из самых распространенных алгоритмов обучения нейросетей прямого распространения является алгоритм обратного распространения ошибки (BackPropagation, BP). Этот алгоритм был переоткрыт и популяризован в 1986 г. Румельхартом и Мак Клелландом из группы по изучению параллельных распределенных процессов в Массачусетском технологическом институте.

Основная идея состоит в том, чтобы вычислять чувствительность ошибки сети к изменениям весов. Для этого нужно вычислить частные производные от ошибки по весам. Пусть обучающее множество состоит из P образцов, и входы k-го образца обозначены через $\{x^i_k\}$. Вычисление частных производных осуществляется по *правилу цепи*: вес входа i-го нейрона, идущего от j-го нейрона, пересчитывается по формуле:

$$\Delta W_{ij} = -\varepsilon * \sum_{k=1}^{P} \frac{\partial E_k}{\partial W_{ij}} = -\varepsilon * \sum_{k=1}^{P} \frac{\partial E_k}{\partial V_k^i} * \frac{\partial V_k^i}{\partial W_{ij}} = -\varepsilon * \sum_{k=1}^{P} \delta_k^i * x_k^i,$$

где ε - длина шага в направлении, обратном к градиенту.

Если рассмотреть отдельно k-ый образец, то соответствующее изменение весов равно:

$$\Delta W_{ij} = -\varepsilon * \frac{\partial E_k}{\partial W_{ij}} = -\varepsilon * \delta_k^i * x_k^i.$$

Множитель δ^i_k вычисляется через аналогичные множители из последующего слоя, и ошибка, таким образом, передается в обратном направлении.

Для выходных элементов получим:

$$\delta_k^i = -\frac{\partial E_k}{\partial v_k^i} = -\frac{\partial E_k}{\partial x_k^i} * \frac{\partial x_k^i}{\partial v_k^i} = -(d_k^i - x_k^i) f'(v_k^i).$$

Для скрытых элементов множитель δ^i_k определяется так:

$$\delta_k^i = -\frac{\partial E_k}{\partial v_k^i} = -\sum_h \frac{\partial E_k}{\partial v_k^h} \frac{\partial v_k^h}{\partial v_k^i},$$

где индекс h пробегает номера всех нейронов, на которые воздействует i-ый нейрон.

Алгоритм обратного распространения ошибки представлен на рисунке 5.

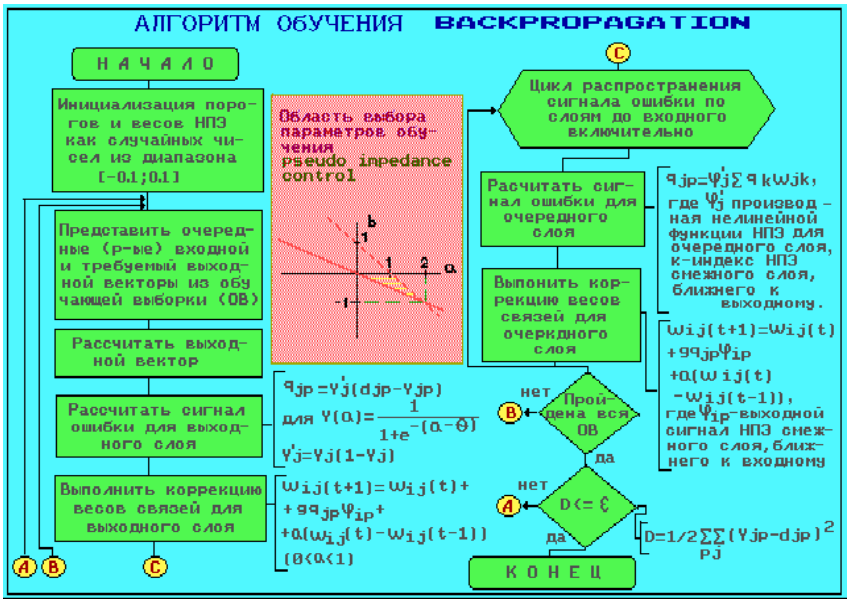


Рис.5 – Алгоритм обучения многослойной нейронной сети

Перед тем, как начинать процесс обучения нейронной сети, необходимо присвоить весам начальные значения. Цель состоит в том, чтобы найти как можно более хорошее начальное приближение к решению и таким образом сэкономить время обучения и улучшить сходимость. Классический подход к этой проблеме состоит в том, чтобы случайным образом выбрать малые значения для всех весов, чтобы быть уверенным, что ни один из сигмоидных элементов не перенасыщен. Однако это не дает полной гарантии, что такое приближение приведет к глобальному минимуму или уменьшит время сходимости. Чтобы обучение не двигалось в ложном направлении при обработке задачи классификации или распознавания, но не задачи аппроксимирования временных рядов, данные нужно перемешивать случайным образом. Иначе нейросеть «выучит» последовательность случайно оказавшихся рядом значений как истинное правило, и потом будет делать ошибку.

Иногда при изменении весов связей нейронов кроме текущего изменения веса к нему прибавляют вектор смещения с

предыдущего шага, взятый с некоторым коэффициентом. В этом случае говорят, что учитывается предыдущий импульс движения. Формула изменения веса связи будет выглядеть следующим образом:

$$\Delta W_{ij}(t+1) = \mu * \Delta W_{ij}(t) - (1-\mu)\varepsilon \frac{\partial E}{\partial W_{ij}}.$$

где μ - число в интервале (0,1), которое задается пользователем.

Меру точности обучения сети определяет «шаг» изменения параметров в процессе управления обучением. Чем он больше, тем более грубым будет следующее уменьшение суммарной ошибки сети. Чем он меньше, тем больше времени сеть будет тратить на обучение и тем более возможно ее попадание в окрестность локального минимума ошибки. Поэтому управление шагом имеет важное значение для улучшения сходимости нейронной сети. В современных нейросетевых пакетах пользователь может сам определять, как будет изменяться величина шага. Очень часто по умолчанию берется линейная или экспоненциальная зависимость величины шага от количества итераций сети.

Одной из самых больших проблем при использовании нейросетей является невозможность предварительного определения оптимального количества скрытых слоев и нейронов в них. Если нейронов будет слишком мало, то это равносильно потере каких-то нелинейных связей в модели, если нейронов будет много, то это может привести к «переобучению» сети, то есть она просто «выучит» данные, а не распознает их структуру. Поэтому применяется два основных подхода:

- деструктивный подход: берется сеть заведомо большего размера, чем нужно, и в процессе обучения из нее удаляются связи и даже сами нейроны;
- конструктивный подход: первоначально берется маленькая сеть, и к ней, в соответствии со структурой и сложностью задачи, добавляются новые элементы.

Диапазон выходных значений решающих функций нейронов лежит в интервале (0,1) либо (-1,1). Поэтому для лучшей работы

сети следует предварительно масштабировать данные обучающей выборки к интервалу от 0 до 1. Это позволит уменьшить ошибки и ускорить функционирование нейросети на этапах обучения и эксплуатации.

Из теоремы об отображении практически любой функции с помощью многослойной нейросети следует, что обучаемая нами нейронная сет **в принципе** способна сама подстроиться под любые данные с целью минимизации суммарной квадратичной ошибки. Чтобы этого не происходило при обучении нейросетей используют следующий способ проверки сети. Для этого обучающую выборку еще перед началом обучения разбивают случайным образом на две подвыборки: обучающую и тестовую. Обучающую выборку используют собственно для процесса обучения, при этом изменяются веса нейронов. А тестовую используют в процессе обучения для проверки на ней суммарной квадратичной ошибки, но при этом не происходит изменение весов. Если нейросеть показывает улучшение аппроксимации и на обучающей, и на тестовой выборках, то обучение сети происходит в правильном направлении. Иначе может снижаться ошибка на обучающей выборке, но происходить ее увеличение на тестовой. Последнее означает, что сеть «переобучилась» и уже не может быть использована для прогнозирования или классификации. В этом случае немного изменяются веса нейронов, чтобы вывести сеть из окрестности локального минимума ошибки.

Самоорганизующиеся карты. Самоорганизующиеся карты (Self Organizing Maps - SOM) это одна из разновидностей нейросетевых алгоритмов. Основным отличием данной технологии от рассмотренных нами ранее нейросетей, обучаемых по алгоритму обратного распространения, является то, что при обучении используется метод обучения без учителя, то есть результат обучения зависит только от структуры входных данных. Нейронные сети денного типа часто применяются для решения самых различных задач, от восстановления пропусков в данных до

анализа данных и поиска закономерностей, например, в финансовой задаче.

Основы самоорганизующихся карт.

Алгоритм функционирования самообучающихся карт представляет собой один из вариантов кластеризации многомерных векторов. Примером таких алгоритмов может служить алгоритм ближайших средних (c-means). Важным отличием алгоритма SOM является то, что в нем все нейроны (узлы, центры классов) упорядочены в некоторую структуру (обычно двумерную сетку). При этом в ходе обучения модифицируется не только нейрон-победитель, но и его соседи, но в меньшей степени. За счет этого SOM можно считать одним из методов проецирования многомерного пространства в пространство с более низкой размерностью При использовании этого алгоритма вектора, схожие в исходном пространстве, оказываются рядом и на полученной карте. SOM подразумевает использование упорядоченной структуры нейронов. Обычно используются одно- и двумерные сетки. При этом каждый нейрон представляет собой n-мерный вектор-столбец w_n , где n определяется размерностью исходного пространства (размерностью входных векторов). Применение одно- и двумерных сеток связано с тем, что возникают проблемы при отображении пространственных структур большей размерности (при этом опять возникают проблемы с понижением размерности до двумерной, представимой на мониторе).

Обычно нейроны располагаются в узлах двумерной сетки с прямоугольными или шестиугольными ячейками. При этом, как было сказано выше, нейроны также взаимодействуют друг с другом. Величина этого взаимодействия определяется расстоянием между нейронами на карте.

При реализации алгоритма SOM заранее задается конфигурация сетки (прямоугольная или шестиугольная), а также количество

нейронов в сети. Некоторые источники рекомендуют использовать максимально возможное количество нейронов в карте. При этом начальный радиус обучения (neighborhood в англоязычной литературе) в значительной степени влияет на способность обобщения при помощи, полученной карты. В случае, когда количество узлов карты превышает количество примеров в обучающей выборке, то успех использования алгоритма в большой степени зависит от подходящего выбора начального радиуса обучения. Однако в случае, когда размер карты составляет десятки тысяч нейронов, время, требуемое на обучение карты, обычно бывает слишком велико для решения практических задач, таким образом, необходимо достигать допустимого компромисса при выборе количества узлов.

Перед началом обучения карты необходимо проинициализировать весовые коэффициенты нейронов. Удачно выбранный способ инициализации может существенно ускорить обучение и привести к получению более качественных результатов. Существуют три способа инициирования начальных весов:

1. инициализация случайными значениями, когда всем весам даются малые случайные величины;
2. инициализация примерами, когда в качестве начальных значений задаются значения случайно выбранных примеров из обучающей выборки;
3. линейная инициализация. В этом случае веса инициируются значениями векторов, линейно упорядоченных вдоль линейного подпространства, проходящего между двумя главными собственными векторами исходного набора данных. Собственные векторы могут быть найдены, например, при помощи процедуры Грама-Шмидта.

Обучение карт состоит из последовательности коррекций векторов, представляющих собой нейроны. На каждом шаге

обучения из исходного набора данных случайно выбирается один из векторов, а затем производится поиск наиболее похожего на него вектора коэффициентов нейронов. При этом выбирается нейрон-победитель, который наиболее похож на вектор входов. Под похожестью в данной задаче понимается расстояние между векторами, обычно вычисляемое в евклидовом пространстве. После того, как найден нейрон-победитель, производится корректировка весов нейро-сети. При этом вектор, описывающий нейрон-победитель, и векторы, описывающие его соседей в сетке, перемещаются в направлении входного вектора.

Использование искусственных нейронных сетей в медицине используется в основном в двух направлениях: синтеза классификационных решающих правил для решения диагностических задач (включая прогнозирование возможных состояний больного при различных терапевтических процедурах и технологиях лечения) и прогнозирования значений регистрируемых показателей, характеризующих функционирование как определенной физиологической или функциональной системы так и организма в целом. Разумеется задача прогнозирования качественно разрешается в экстраполяционном временном интервале.

Для решения задачи классификации используются различные пакеты программного обеспечения: Statistica, Mahlab Simulink. Neirosolution, специализированные средства для решения определенных задач.

Рассмотрим примеры синтеза нейронных сетей в универсальных оболочках.

1. *В пакете STATISTICA:*

Для решения задач классификации используется *Мастер решения задач.*

Первый шаг: открывается подготовленный файл данных; ряд переменных (столбцов) являются значениями параметров, категорильная переменная (последний столбец) обозначает диагностический класс.

Второй шаг: открывается окно «*Тип задачи» («Problem Type»)*; указывается тип задачи и нажимается кнопка «*Next*».

Третий шаг: в окне выбирается зависимая переменная (диагностического класса) и нажимается кнопка «*Next*».

Четвертый шаг: в окне выбираются входные (независимые) переменные; после нажатия кнопки «*Next*» *Мастер решения* автоматически разобьет выборку на обучающую, контрольную и тестовую подвыборки, выделив их черным, синим и красным цветом, соответственно, и производит перемешивание наблюдений для обеспечения репрезентативности подвыборок.

Пятый шаг: на экране появляется окно «*Длительность поиска*» («Duration of Design Process»), в котором задается длительность поиска: быстрый, средний, полный, ограниченный по времени.

Шестой шаг: в открывшемся окне «*Сохранение сетей» («Saving Networks»)*определяется способ сохранения сетей, например: сохранить сети с лучшим качеством решения, максимальное число сохраняемых сетей и т.д.; затем, в следующем открывшемся окне указываются опции представления результатов.

Седьмой шаг: после нажатия кнопки «*Finish*» STATISTICA производит вычисления и предоставляет итоговый результат в виде таблицы - например, если был выбран на предыдущем шаге режим «сохранить 10 сетей с лучшим качеством», то на экране появятся следующие столбцы: – *Type* – указывается тип сетей, *RBF* – указываются радиальные базисные функции, *Linear* – линейные, *MLP* – многослойный персептрон, ошибка, входы, скрытые, *Perfomance* – качество (указаны доли правильно классифицированных диагностических состояний) – определяется по контрольному подмножеству (доля правильно классифицированных наблюдений – диагностическая эффективность).

Лучшая сеть отмечается знаком «*». Для удаления из набора сетей неудовлетворительного качества используется правая кнопка мыши и команда «*Удалить*» («*Delete*»).

Выделенная нейронная сеть делается активной с помощью команды всплывающего меню «*Выбрать*» («*Select*»).

Если набор нейронных сетей заполнен, то программа *ST Neural Networks* должна определить, какие из имеющихся сетей заменяются вновь создаваемые. После нажатия кнопки *Options-Опции* в диалоговом окне *Редактор набора сетей – Network Set Editor*. На экране появиться окно *Параметры набора сетей – Network Set Options*, в котором задается максимальное количество сетей в наборе (по умолчанию – 30). Установив необходимые значения параметров набора сетей нажимается кнопка *Закрыть – Close*.

Для исследования степени информативности входных переменных в обученной сети применяют *анализ чувствительности* – для этого используют команду *Чувствительность – Sensitiivity* из выпадающего меню. Программа строит таблицу, в которой указывается чувствительность сети по отношению к каждой переменной: ранг, ошибка и отношение. Показатели чувствительности определяются отдельно для обучающего и контрольного набора (подвыборки) наблюдений. В качестве столбцов в окне выступают переменные исходного файла данных.

2. Классификация с помощью персептрона в пакете MatLab.

Классификация состоит из этапов: ввод данных, обучение сети, классификация. Для оценки качества классификации (диагностики) первоначально результаты обследования (и-или результаты мониторинга за состоянием) биообъекта разделяются на две подвыборки: обучающая и тестовая. Если руководствоваться принципом «золотого сечения», то соотношение размеров указанных выборок -0,62:0,38.

Рассмотрим пример 1, реализованный в пакете MatLab 7 SP 2 + Simulink 6, для обучения **персептрона**:

```
>> % одному из двух классов
>> P=[-0.5 -0.5 +0.3 – 0.1;-0.5 +0.5 -0.5 +1.0];
>> T=[1 1 0 0];
>> plot (P,T);
>> % графическое представление исходных данных
>> %  создание персептрона
>> % с указанием границ изменений и одним нейроном
>> My_net=new([1 1; -1 1], 1); % инициализация персептрона
>> %  организация цикла адаптивной настройки персептрона
```

```
>> % с показом графика линии разделения классов
>> %  while (sse(E))
        [My_net, Y,E]=adapt(My_net, P, T);
        Linehandle=plotpc(My_net.IW{1}, My_net.b{1});
        Drawnow;
        End;
```

В приведенной программе исходными данными являются входные векторы с указанием их принадлежности к одному из двух классов – с индексами 0 и 1. P – матрица, столбцы которой соответствуют 4-м входным векторам, T – вектор, элементы которого указывают на принадлежность вектора к нулевому или первому классу. Итоговый результат отображает обучающие вектора и разделяющую их линию, формируемую обученным персептроном.

Для проверки качества созданного классификатора подадим, например, на вход обученного персептрона контрольный вектор p={0.4; 0.5} и проанализируем реакцию сети. Моделирование иллюстрирует следующий программный фрагмент.

```
>> p=[0.4; 0.5];
>> = = sim(My-net,p);
>>a
a=
0
```

Его работа показывает, что последовала реакция сети a=0, т.е., предъявленный контрольный вектор относится к нулевому классу, что является правильным (и графически подтверждается рисунком).

Построение графика реализуется следующим программным фрагментом:

```
>>  plot(p,a);
>> hold on;
>> plotpv(P,T);
>>  Linehandle=plotpc(My_net.IW{1}, My_net.b{1});
>> hold off
```

Для оценки диагностического качества здесь и далее с помощью статистических вычислений ошибок первого и второго рода классификации необходимо на контрольной выборке для ряда объектов выполнить анализ правильности срабатывания

персептрона и рассчитать значения показателей диагностических чувствительности, специфичности и эффективности.

Рассмотрим пример 2, реализованный в пакете MatLab 7 SP 2 + Simulink 6, для обучения **сети со слоем Кохонена**:

*Задание***:** Используя встроенные функции пакета нейронных сетей математической среды Matlab, построить нейронную сеть со слоем Кохонена, которая множество входных данных разделит на кластеры и выявит их центры. На обученную сеть подать новый входной вектор и определить, к какому кластеру он относится.

Для создания нейронной сети со слоем Кохонена воспользуемся встроенной в среду Matlab функцией *newc:*

```
X=[0 1; 0 1];
clusters=5;
points=5; %3адание количества точек в кластере
std_dev=0.01;
p=nngenc(X,clusters,points,std_dev);%Моделирование входных данных
h=newc([0 1;0 1],5,.1); % создание слоя Кохонена
h.trainParam.epochs=50; %3адание количества циклов обучения
h=init(h);
h=train(h,p);
w=h.IW{1}; % вывод графиков исходных данных и выявленных центров кластеров
plot(p(1,:),p(2,:),'^r'),grid;
hold on;
plot(w(:,1),w(:,2),'ob');
xlabel('p(1)');
ylabel('p(2)');
A=0.6
B=0.5
p=[A;B];
plot(A,B,'^k');
y=sim(h,p) %Опрос сети
A =0.6000
B =0.5000
y = (2,1)
```

Результат работы программы отображается в командном окне.

Предъявленный вектор отнесен ко второму кластеру. Теперь данный алгоритм применим к реальной (хоть и элементарной) задаче кластеризации. На вход нейронной сети будем подавать данные весоростовых показателей людей и попробуем выявить три

класса (кластера): нормальный весоростовой показатель; избыток веса; недостаток веса. Программный код выглядит следующим образом:

```
%входные данные (первая строка матрицы - рост; вторая - вес)
p=[175 180 182 175 183 176 183 176 183 176 175 180 178 180 178 182 178 182 179 174 172 179; 70 75 100 99 42 48 76 72 40 45 92 96 70 69 95 90 79 82 80 50 96 91]  %создаем НС Кохонена с 3 кластерами (нормальный весоростовой показатель, избыток веса и недостаток веса)
h=newc([0 200;0 100],3,.1);
h.trainParam.epochs=500; %Задание количества циклов обучения
h=train(h,p);
w=h.IW{1};
plot(p(1,:),p(2,:),'^r');
hold on;
plot(w(:,1),w(:,2),'ob');
xlabel('Rost');
ylabel('Ves');
% Задание нового входного вектора
%Опрос сети
A=181
B=65
p=[A;B];
plot(A,B,'+r')
y=sim(h,p)
A =181
B = 65
y = (2,1) 1
```

Результат работы программы представлен в командном окне: *y=(2,1) 1*. Предъявленный вектор отнесен ко второму кластеру.

3. Теперь рассмотрим использование ***самоорганизующей карты*** на примере двумерных векторов. Используя самоорганизующиеся карты, двумерные векторы разбить на кластеры и выявить их центры, затем подать на вход самоорганизующей карты новый вектор и определить кластер, к которому он относится.

```
P=rands(2,100); %Задание случайных двухмерных входных векторов
figure(1);
hold on;
plot(P(1,:),P(2,:),'+r') %визуальное изображение входных векторов
%Создание НС с 3*4 нейронами
%По умолчанию функция TFCN = 'hextop', то есть нейроны располагаются в узлах двумерной сетки с шестиугольными ячейками
net=newsom([0 1;0 1],[3 4]);
```

```
net.trainParam.epoch=1 %3адание числа циклов настройки
net=train(net,P); % настройка сети
A=0.5
B=0.3
p=[A;B]; % Задание нового входного вектора
plot(A,B,'^k') %прорисовка на рисунке входного вектора (черный треугольник)
figure(2)
plotsom(net.iw{1,1},net.layers{1}.distances)
a=sim(net,p) %опрос сети
A = 0.5000
B = 0.3000
a = (5,1) 1
```

4. Результат работы программы представлен в командном окне: a = (5,1) 1/ Предъявленный вектор отнесен к пятому кластеру.

Порядок выполнения практической работы:

1. Изучить теоретический материал и принципы построения искусственных нейронных сетей, включая язык программирования в MatLab 7 SP 2 + Simulink 6.
2. Сформировать выборки исходных данных по Таблице 1, характеризующие результаты общего анализа крови. Выборки состоят из:
- для пакета STATISTICA - матрицы, включающей три показателя крови (согласно варианта задания – определяется преподавателем) и идентификаторов состояния («здоров», «болен»), трех матриц типа «показатель», «состояние»;
- для MatLab 7 SP 2 + Simulink 6 - трех матриц типа «показатель», «состояние».
3. Построить нейронные сети в пакетах STATISTICA и MatLab 7 SP 2 + Simulink 6 (или иных версиях MatLab) – персептрон, сеть Кохонена, самоорганизующие карты для всех сформированных в п.2 матриц.
4. По результатам п.3 сравнить качество работы сетей – в качестве показателя качества использовать диагностическую эффективность.
5. По зафиксированным скрин-шотам работы с программными инструментариями (представляются в отчете по выполнению лабораторной работы) и результатами п.4 сделать выводы о

перспективности применения полученных сетей для диагностического процесса.
6. Оформить отчет.

Контрольные вопросы:
1. Что понимается под кластеризацией?
2. Для чего применяются искусственные нейронные сети?
3. Каким образом классифицируются искусственные нейронные сети?
4. Для решения каких задач предназначены искусственные нейронные сети?
5. В чем заключается закон обучения Кохонена?
6. Опишите встроенные операторы Matlab для кластеризации.
7. Зачем используются самоорганизующиеся карты? Чем отличаются сети Кохонена от SOM?
8. Как устроен персептрон?
9. В чем заключается принцип обратного распространения ошибки в искусственных нейронных сетях?
10. Что такое – «скрытые слои» искусственной нейронной сети?
11. Как проверяется диагностическое качество работы искусственной нейронной сети?
12. Перечислите основные недостатки и преимущества применения искусственных нейронных сетей в диагностическом процессе?
13. В чем состоят отличия искусственных нейронных и иммунных сетей?

Библиография.
1. Боровиков, В. STATISTICA. Искусство анализа данных на компьютере [Текст] / В. Боровиков. - 4-е изд. – СПб. : Питер, 2022. - 688
2. Искусственные нейронные сети [Электронный ресурс]: методические указания к выполнению лабораторной работы по дисциплине «Использование программы MathLab в инженерной практике» для магистров направления подготовки 211000.68 «Конструирование и технология электронных средств» / Юго-

Запад. гос. ун-т ; сост. Е. О. Брежнева. - Электрон. текстовые дан. (569 КБ). - Курск : ЮЗГУ, 2014. - 17 с
3. Искусственные нейронные сети. Презентация / http://mrmarker.ru/p/page.php?id=6218
4. Искусственные нейронные сети. Презентация. / http://5informatika.net/informatsionnye-sistemy/Nejronnye-seti.html
5. Классификация информативных признаков с помощью нейронной сети Кохонена [Электронный ресурс]: методические указания для проведения лабораторных работ и выполнения самостоятельной внеаудиторной работы по дисциплине «Нейрокомпьютерные системы» для студентов направления подготовки бакалавров 09.03.04 «Программная инженерия» / Юго-Зап. гос. ун-т ; сост. Р. А. Томакова. - Электрон. текстовые дан. (534 КБ). - Курск : ЮЗГУ, 2016. - 16 с
6. Кохонен Т. Самоорганизующие карты .-М.: БИНОМ. Лаборатория знаний, 2016. – 655 с.
7. Нейронные сети: история развития теории [Текст]: учебное пособие: Кн. 5 / Под общ. ред. А. И. Галушкина, Я. З. Цыпкина. - М.: ИПРЖ, 2001. - 840 с.
8. Математические методы информатики в задачах и примерах. Опыт применения в проектировании сложных систем [Текст] : учебное пособие / под ред. Ю. П. Мухи, В. И. Сырямкина ; Национальный исследовательский Томский политехнический университет. - Томск : Изд-во Том. ун-та, 2012. - 484 с.
9. Основы нейрокибернетики, генетические алгоритмы [Текст] : учебное пособие для студентов по специальности 230700, 230400, 010500, 060101, 060109 / Министерство образования и науки РФ, ФГБОУ ВПО "Юго-Западный университет" ; ЮЗГУ, ТулГУ ; сост. : О. Г. Павлов, Ю. А. Халин. - Тула : ТулГУ, 2014. - 103 с
10. Реализация нейронных сетей с помощью пакета NNTool системы MATLAB [Электронный ресурс] : методические указания для проведения лабораторных работ и выполнения самостоятельной внеаудиторной работы по дисциплине «Нейрокомпьютерные системы» для студентов направления подготовки бакалавров 09.03.04 «Программная инженерия» / Юго-

Зап. гос. ун-т ; сост. Р. А. Томакова. - Электрон. текстовые дан. (564 КБ). - Курск : ЮЗГУ, 2016. - 17 с.
11. Романов, В. П. Анализ финансовых данных с помощью самоорганизующихся карт на базе нейропакета "Statistica neural networks" [Текст] : лабораторная работа по дисциплине "АИС по областям применения" / под общ. ред. К. И. Курбакова ; Министерство образования и науки Российской Федерации, Российская экономическая академия им. Г. В. Плеханова. - М. : КОС-ИНФ, 2006. - 32 с.
12. Системы искусственного интеллекта. Практический курс [Текст] : учебное пособие / ред. И. Ф. Астахова. - М. : БИНОМ.Лаборатория знаний, 2008. - 292 с
13. Элементарное введение в технологию нейронных сетей с примерами программ [Текст] / [пер. с пол. И. Д. Рудинского]. - М. : Горячая линия - Телеком, 2011. - 408 с

Приложение

Таблица 1. Данные для построения искусственной нейронной сети.

Больные					
№ п.п.	эритроциты	гемоглобин	Цветовой показатель	моноциты	СОЭ
1	4,6	142	0,93	2	9
2	4,2	115	0,82	2	16
3	3,4	107	0,94	2	8
4	5	170	1,02	4	2
5	3,6	105	0,88	2	22
6	5,3	108	0,61	2	6
7	4,16	132	0,95	4	10
8	4,3	145	1,01	3	4
9	5	160	0,96	1	4
10	3,1	108	1,05	4	12
11	4	124	0,93	3	10
12	5,1	164	0,96	6	12
13	4,25	138	0,97	4	26
14	4,3	150	1,05	9	9
15	4,8	153	0,96	2	9
16	3,6	114	0,95	1	14
Здоровые					
1	4,2	137	0,98	3	3
2	4,05	125	0,93	2	7
3	3,9	125	0,96	9	5
4	3,85	120	0,94	5	5
5	4,2	134	0,96	8	4
6	3,65	117	0,96	3	9
7	4,4	146	0,995	4	3
8	4	122	0,92	2	10
9	4,7	150	0,96	3	6
10	4,6	144	0,94	6	3
11	4,26	140	0,99	6	3
12	4,2	140	1	6	6
13	4,5	140	0,93	9	1
14	4,5	150	1	4	4
15	4,4	148	1,01	10	2
16	4,4	143	0,98	8	4

12. ПРИМЕНЕНИЕ ДИСКРИМИНАНТНОГО АНАЛИЗА ДЛЯ ПОСТРОЕНИЯ РЕШАЮЩИХ ПРАВИЛ В КЛАССИФИКАЦИОННЫХ КОМПОНЕНТАХ СИСТЕМ УПРАВЛЕНИЯ.

Цель работы: изучение возможностей о овладение навыками применения дискриминантного анализа при решении классификационных задач.

Краткие теоретические сведения

В медицине часто возникают задачи соотнесения неизвестного биообъекта или процесса устройства к определенной нозологической группе, классификации исследуемого объекта по результатам применения информационно-аналитической техники, соотнесения состояния к определенной стадии функционирования (например, его дальнейшей пригодности). В простейшем случае, эти задачи могут быть решены с помощью линейного дискриминантного анализа.

Дискриминантный анализ используется для принятия решения о том, какие переменные различают (дискриминируют) две или более возникающие совокупности (группы, классы). 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). Специалист в области приборостроения может, анализируя характеристики работоспособности, соотнести состояние прибора к определенному этапу его жизненного цикла.

С вычислительной точки зрения дискриминантный сходен с дисперсионным анализом. *Цель дискриминантного анализа заключается* в определении, отличаются ли совокупности по среднему какой-либо переменной (или линейной комбинации переменных), и затем использовать эту переменную, чтобы осуществлять прогноз для новых объектов исследования их принадлежность к определенной

Таким образом, задача о дискриминантной функции может быть перефразирована как задача одно входового дисперсионного

анализа (ANOVA). В случае одной переменной окончательный критерий значимости того, разделяет переменная две совокупности или нет, дает *F*-критерий.

При применении дискриминантного анализа имеются несколько переменных, и задача состоит в том, чтобы установить, какие из переменных вносят свой вклад в дискриминацию между совокупностями. Наиболее общим применением дискриминантного анализа является включение в исследование многих переменных с целью определения тех из них, которые наилучшим образом разделяют совокупности между собой. В пошаговом анализе дискриминантных функций модель дискриминации на каждом шаге просматриваются все переменные и находится та из них, которая вносит наибольший вклад в различие между совокупностями. Эта переменная включается в модель на данном шаге, и происходит переход к следующему шагу. В процессе идентификации дискриминантной функции можно двигаться и в «обратном направлении». В этом случае, все переменные включаются в модель, а затем на каждом шаге устраняются переменные, вносящие малый вклад в предсказания.

Интерпретация функций дискриминации для двух классов осуществляется следующим образом. Для двух классов дискриминантный анализ может рассматриваться как процедура множественной регрессии (и аналогичная ей). Интерпретация результатов задачи с двумя совокупностями тесно следует логике применения множественной регрессии: переменные с наибольшими регрессионными коэффициентами вносят наибольший вклад в дискриминацию. При построении дискриминантных функций для более чем 2 классов используются подход постепенного бинарного деления групп. (Заметим, что это деление носит субъективный характер, то возможно возникновение ситуации синтеза различных функций, в зависимости от последовательности бинарного разделения на классы с близкими значениями критерия качества классификации.)

Полученные значения структурных коэффициентов применяются и при интерпретации «реального смысла» дискриминирующей функции.

Значимость дискриминантной функции оценивается путем проверки числа корней, которое значимо добавляется к дискриминации между совокупностями. Для интерпретации могут быть использованы только те из них, которые будут признаны статистически значимыми. Остальные функции (корни) должны быть проигнорированы.

Таким образом, при интерпретации дискриминантной функции для нескольких совокупностей и нескольких переменных, вначале хотят проверить значимость различных функций и в дальнейшем использовать только значимые функции. Затем, для каждой значащей функции рассматриваются стандартизованные коэффициенты *бета*. Чем больше стандартизованный коэффициент *бета,* тем большим является относительный собственный вклад переменной в дискриминацию, выполняемую соответствующей дискриминантной функцией.

Дискриминантный анализ позволяет получить качественный результат при выполнений следующих требований:

1. Анализируемые переменные представляют выборку из многомерного нормального распределения. Для проверки этого требования используют специальные критерии и графики.
2. Матрицы дисперсий/ковариаций переменных однородны.
3. Применяемые критерии оценки значимости дискриминантной функции достаточно корректны. Большинство «реальных угроз» корректности применения критериев значимости возникает из-за возможной зависимости между средними по совокупностям и дисперсиями между собой. Интуитивно ясно, что если имеется большая изменчивость в совокупности с высокими средними в нескольких переменных, то эти высокие средние ненадежны. Однако критерии значимости основываются на объединенных дисперсиях, то есть, на средней дисперсии по всем совокупностям. Поэтому критерии значимости для относительно больших средних (с большими дисперсиями) основываются на относительно меньших объединенных дисперсиях и будут ошибочно указывать на статистическую значимость. Практически это происходит, если одна из изучаемых совокупностей содержит несколько экстремальных выбросов, которые сильно влияют на средние и,

таким образом, увеличивают изменчивость. Для определения такого случая следует изучить описательные статистики, то есть средние и стандартные отклонения или дисперсии для таких корреляций.

Переменные, используемые для дискриминации между совокупностями, не являются полностью избыточными. При вычислении результатов дискриминантного анализа происходит обращение матрицы дисперсий/ковариаций для переменных в модели. Если одна из переменных полностью избыточна по отношению к другим переменным, то такая матрица называется *плохо обусловленной* и не может быть обращена. Например, если переменная является суммой трех других переменных, то это отразится также и в модели, и рассматриваемая матрица будет плохо обусловленной.

Чтобы избежать плохой обусловленности матриц, необходимо постоянно проверять так называемые *значения толерантности* для каждой переменной. Значение толерантности вычисляется как $(1-R^2)$, где R^2 - квадрат коэффициент множественной корреляции для соответствующей переменной со всеми другими переменными в текущей модели (известный так же под названием – коэффициент детерминации).

Основной целью применения дискриминантного анализа является проведение классификации анализируемого объекта. Как только модель установлена и получены дискриминирующие функции, возникает вопрос о том, как хорошо они могут *предсказывать*, к какой совокупности принадлежит конкретный объект исследования? Между тем: функции классификации не следует путать с дискриминирующими функциями. Функции классификации предназначены для определения того, к какой группе наиболее вероятно может быть отнесен каждый объект. Имеется столько же функций классификации, сколько групп. Каждая функция позволяет вам для каждого образца и для каждой совокупности вычислить веса классификации по формуле:

$$S_i = c_i + w_{i,1} \cdot x_1 + w_{i,2} \cdot x_2 + ... + w_{i,m} \cdot x_m ,$$

где индекс i обозначает соответствующую совокупность, а индексы 1, 2, ..., m обозначают m переменных; c_i являются константами для i-ой совокупности, $w_{i,j}$ - веса для j-ой переменной при вычислении показателя классификации для i-ой совокупности; x_j - наблюдаемое значение для соответствующего образца j-ой переменной. Величина S_i является результатом.

В общем случае наблюдение считается принадлежащим тому классу (группе) совокупности, для которой получен наивысший показатель классификации (кроме случая, когда вероятности априорной классификации становятся слишком малыми ниже). Для определения апостериорной вероятности правильной классификации используются понятие расстояния Махаланобиса.

Расстояние Махаланобиса. Вы можете прочитать об этих расстояниях в других разделах. В общем, расстояние Махаланобиса является мерой расстояния между двумя точками в пространстве, определяемым двумя или более *коррелированными* переменными. Например, если имеются всего две некоррелированных переменные, то вы можете нанести точки (образцы) на стандартную двухмерную диаграмму рассеяния. 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. Снова, 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. Это значение будет не вполне точным, так как распределение вокруг среднего для каждой совокупности будет не в точности нормальным. Так как принадлежность каждого образца вычисляется по априорному знанию модельных переменных, эти вероятности называются *апостериорными* вероятностями и вычисляются с использованием знания значений других переменных для образцов из частной совокупности. Некоторые пакеты автоматически вычисляют эти вероятности для всех наблюдений (или для выбранных наблюдений при проведении кросс-проверки).

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.

Итог классификации. Общим результатом, на который следует обратить внимание при оценке качества текущей функции классификации, является *матрица классификации*. Матрица классификации содержит число образцов, корректно классифицированных (на диагонали матрицы) и тех, которые попали не в свои совокупности (группы).

Порядок выполнения.

1. Сформируйте обучающие и экзаменационные выборки на материале Приложения. Закодируйте состояния кластеров заболеваний любым образом (например, коды «-10», «0», «10»).
2. Получите на обучающей выборке линейную дискриминантною функцию в Excel с помощью множественного линейного регрессионного анализа, считая коду выходной функцией. Оцените эффективность диагностического правила на экзаменационной выборке.
3. Получите на обучающей выборке линейную дискриминантною функцию в Statistica. Оцените эффективность диагностического правила на экзаменационной выборке.
4. Сравните результаты, полученные в п.2 и 3.

Примечание: эффективность определяется как отношение правильно диагностируемых объектов во всех классах к общему количеству объектов – вычисляется на экзаменационной выборке)

Контрольные вопросы.

1. В каких случаях возникает проблема решения задач классификации в кибернетических системах?
2. Каким образом осуществляется проверка качества диагностических решающих правил?
3. Основные задачи, которые решают кластерный и дискриминантный анализ?
4. Зачем нужна проверка на толерантность?
5. Что такое тестовая выборка?
6. Какие существуют вероятности принадлежности отдельных значений выбранной переменной к той или иной группе?
7. Что такое функция классификации?
8. Что такое кросс-проверка?
9. Какие существуют меры расстояний между объектами при древовидной кластеризации?
10. Правила объединения или связи для двух кластеров?
11. Зачем программа изменяет принадлежность объектов к кластерам в методе К средних?
12. В чем заключается сущность метода К средних?
13. Какие стандартные пакеты статистического исследования позволяют проводить дискриминантный анализ?
14. В чем сущность нелинейного дискриминантного анализа?
15. Какие электронные устройства (узлы, элементы) позволяют решать задачи дискриминантного анализа?

Библиография

1. Воронцов И.М., Шаповалов В.В., Шерстюк Ю.М. Здоровье. Опыт разработки и обоснование применения автоматизированных систем для мониторинга и скринирующей диагностики нарушений здоровья. – СПб.: ООО «ИПК «Коста»Б, 2006. – 432 с.
2. Дискриминантный анализ. Презентация / http://www.myshared.ru/slide/398815/
3. Емельянов С. Г. Встраиваемые оптико-электронные устройства распознавания изображений в многомерном пространстве признаков [Текст] : учебное пособие / С. Г. Емельянов, Д. В. Титов ; Юго-Зап. гос. ун-т. - Курск : ЮЗГУ, 2013. - 130 с
4. Дискриминантный анализ. Электронный учебник по статистике / /http://www.statsoft.ru/home/textbook/modules/stdiscan.html

Приложение

эр	Hb	ЦП	лейк	эоз	пал	сегм	лимф	мон	заболевание
4,6	142	0,93	7,05	1	14	45	38	2	рак желудка
4,2	115	0,82	7,8	2	4	81	10	2	рак желудка
3,4	107	0,94	8	1	6	64	27	2	рак желудка
3	87	0,87	7,3	4	1	55	36	4	рак желудка
3,4	100	0,88	6,15	0	4	77	17	2	рак желудка
5,0	170	1,02	7,1	2	5	59	30	4	рак желудка
5	150	0,90	3,7	6	1	71	21	1	рак желудка
3,6	105	0,88	17,6	0	7	64	26	2	рак желудка
5,3	108	0,61	4,2	4	1	47	46	2	рак желудка
4,16	132	0,95	5,1	2	4	43	47	4	рак желудка
4,3	145	1,01	4,55	4	5	52	36	3	рак желудка
4,0	128	0,96	7,6	4	2	57	29	7	рак желудка
5,0	160	0,96	6,8	0	4	72	23	1	рак желудка
4,46	146	0,98	6,5	4	1	66	27	2	рак желудка
5,4	176	0,98	7,8	2	2	67	29	0	рак желудка
3,1	108	1,05	3,7	1	2	62	31	4	рак желудка
4,0	124	0,93	4,4	2	7	66	22	3	рак желудка
5,1	164	0,96	7,9	0	5	29	60	6	рак желудка
4,25	138	0,97	6,2	2	7	59	28	4	рак желудка
4,8	160	1,00	6,7	2	2	70	25	1	рак желудка
4,3	150	1,05	12	1	1	59	30	9	рак желудка
4,0	113	0,85	5,85	6	4	48	37	5	рак желудка
4,8	153	0,96	7,1	6	1	61	30	2	рак желудка
3,6	114	0,95	13,1	0	2	73	24	1	рак желудка
3,35	103	0,92	9,1	1	3	58	27	17	рак желудка
4,5	150	1,00	8	4	1	56	36	3	рак желудка
эр	Hb	ЦП	лейк	эоз	пал	сегм	лимф	мон	
4,6	130	0,85	6,7	1	3	44	38	14	Цирроз
3,7	117	0,95	3,6	5	3	50	32	10	цирроз
3,3	110	1,00	4,4	1	3	70	23	3	цирроз
4,4	140	0,95	13,2	3	2	63	30	2	цирроз
4,5	150	1,00	7	1	6	50	34	9	цирроз
4,4	132	0,90	4,4	0	5	60	33	2	цирроз
4,1	136	1,00	5	1	1	70	24	4	цирроз
4	130	0,98	4,9	3	2	71	20	4	цирроз
3,5	115	0,99	7,8	1	6	51	38	4	цирроз
4,2	132	0,94	6	3	4	67	20	6	цирроз
4,3	140	0,98	4,3	2	2	71	17	8	цирроз
4	118	0,89	8,3	0	5	61	26	8	цирроз
4,5	144	0,96	6,4	2	1	60	33	4	цирроз

4,60	150	0,98	5,6	7	4	46	36	7	цирроз
4	130	0,98	7,0	5	6	54	26	9	цирроз
3,9	130	1,00	9,6	1	19	48	15	5	цирроз
4,0	112	0,84	15,8	4	18	61	13	4	цирроз
4,6	150	0,98	5,4	1	2	52	37	8	цирроз
4,6	150	0,98	5,4	2	5	51	34	8	цирроз
4,6	145	0,95	5,2	1	8	68	20	3	цирроз
4,6	150	0,98	6,6	2	3	64	42	12	цирроз
3,2	108	1,01	3	2	5	48	49	6	цирроз
4,5	141	0,94	8,4	2	4	58	24	12	цирроз
4,2	140	1,00	7,1	1	3	60	33	3	цирроз
4	130	0,98	7,0	5	6	54	26	9	цирроз
3,7	112	0,91	3,9	2	1	61	32	4	цирроз
4,0	130	0,98	4	3	1	57	35	4	цирроз
3,7	114	0,92	3,8	3	2	50	41	4	цирроз
4,4	144	0,98	7	2	12	48	26	12	цирроз
эр	Hb	ЦП	лейк	эоз	пал	сегм	лимф	мон	
3,9	109	0,84	4,6	3	3	67	24	3	желчКамБол
4,0	132	0,99	7	3	5	84	10	1	желчКамБол
4,4	144	0,98	9,1	4	7	72	20	1	желчКамБол
3,6	126	1,05	4,4	2	8	54	37	1	желчКамБол
3,5	106	0,91	11,9	1	4	73	18	5	желчКамБол
3,0	100	1,00	6,0	3	1	60	30	6	желчКамБол
3,9	139	1,07	6,4	2	4	59	34	1	желчКамБол
3,8	127	1,00	5,9	3	4	40	50	6	желчКамБол
3,1	103	1,00	10,2	4	1	68	17	10	желчКамБол
3,8	120	0,95	6,8	1	1	58	37	3	желчКамБол
4,1	134	0,98	8,9	2	5	58	36	1	желчКамБол
4,0	120	0,90	7,2	3	1	78	19	2	желчКамБол
3,5	126	1,08	5,3	1	1	51	46	2	желчКамБол
3,5	114	0,98	5,2	3	3	59	33	2	желчКамБол
3,9	126	0,97	9,3	1	11	66	20	2	желчКамБол
4,35	138	0,95	19,0	2	8	86	3	1	желчКамБол
3,8	124	0,98	7,2	2	7	69	27	2	желчКамБол
3,8	128	1,01	10,2	1	8	77	14	1	желчКамБол
3,5	110	0,94	9,4	3	6	68	24	2	желчКамБол
3,5	114	0,98	14,9	2	3	79	15	3	желчКамБол
3,9	126	0,97	9,9	4	5	76	18	1	желчКамБол
3,9	120	0,92	17,9	3	5	67	24	4	желчКамБол
4,5	140	0,93	12,7	1	8	87	3	1	желчКамБол
3,3	110	1,00	3,2	2	1	66	27	4	желчКамБол