

**Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Владимирский государственный университет имени
Александра Григорьевича и Николая Григорьевича Столетовых»
(ВлГУ)**

Институт инновационных технологий

Факультет информационных технологий

Кафедра Информационных систем и программной инженерии

Р.И. МАКАРОВ

МОДЕЛИРОВАНИЕ

*Методические указания
к лабораторным занятиям*

Учебное электронное издание

Владимир 2013

УДК 004 (076)
ББК 32.97я7. - ISBN

M15

Методические указания к лабораторным занятиям по дисциплине «Моделирование» предназначены для подготовки магистров по дисциплине профессионального цикла по направлению подготовки - 231000-Программная инженерия, программе подготовки – “Разработка программно-информационных систем”.

Курс знакомит магистрантов с методами построения моделей сложных систем, с возможностями средств моделирования, оценкой качества моделей, проведением экспериментов для оценки эффективности сложных систем, применением моделей в задачах управления.

Слушатели приобретают практические навыки использования методов моделирования сложных систем для решения задач в области информационных систем и информационных технологий.

Лабораторные занятия знакомят магистрантов с методами разработки математических и имитационных моделей, с технологией подготовки и проведения имитационного моделирования систем; с разработкой моделей предметных областей информационных систем.

© Владимирский государственный университет, 2013

Лабораторная работа №1

Моделирование процесса обслуживания заявок в информационной системе

1. Цель работы

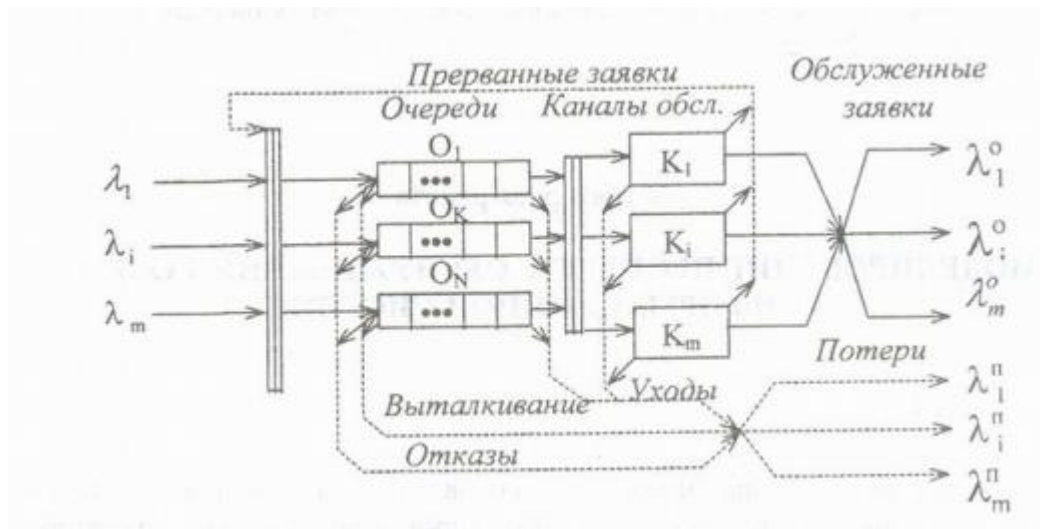
Цель работы – ознакомление магистрантов с использованием элементов теории массового обслуживания для моделирования и оценки параметров информационной системы.

2. Общие сведения

Элементы теории массового обслуживания

Для современных вычислительных машин и систем характерна работа в режиме решения потока случайных по своим характеристикам задач, поступающих в общем случае в случайные моменты времени. Анализ и, самое главное, синтез подобных систем с учётом вероятностного характера протекающих в них процессов возможны с использованием методов теории массового обслуживания.

Предмет теории массового обслуживания – системы и сети массового обслуживания. Под системой массового обслуживания (СМО) понимают динамическую систему, предназначенную для эффективного обслуживания случайного потока заявок (требований на обслуживание) при ограничениях на ресурсы системы. Обобщённая структурная схема приведена на рисунке.



Поступающие на вход СМО однородные (то есть требующие однородного обслуживания) заявки в зависимости от порождающей их причины делятся на типы, интенсивность потока заявок типа i ($i = \overline{1, M}$) обозначается как λ_i . Первопричина заявок, какова бы ни была её физическая природа, называется источником заявок, совокупность заявок всех типов – входящим потоком СМО.

Обслуживание заявок выполняется совокупностью m в общем случае разнотипных каналов. В произвольный момент времени канал может быть занят обслуживанием только

одной заявки, в общем случае допускается прерывание начатого некоторым каналом процесса обслуживания.

Если в момент появления заявки на входе СМО хотя бы один канал свободен, её обслуживание может быть начато немедленно, без задержки. Однако, вполне вероятна ситуация, когда заявка застаёт СМО полностью загруженной, то есть когда все m каналов заняты обслуживанием. В этом случае начало обслуживания задерживается, заявка должна занять место в соответствующей очереди. Очередь может быть либо общей, либо отдельной; деление очереди выполняется обычно по приоритетному принципу. На число мест в очереди может быть наложено ограничение, это может быть сделано как для каждой очереди в отдельности, так и для всей совокупности очередей в целом. При этом возможны конфликтные ситуации, решением которых может быть либо отказ системы принять заявку, либо принятие заявки за счёт возможно, выталкивания из очереди другой, менее ценной заявки в данный момент времени.

В зависимости от числа мест в очереди различают СМО с отказами и без отказов. В СМО с отказами число мест в очереди конечно.

В зависимости от допустимого времени пребывания заявки в системе различают СМО с «нетерпеливыми» и «терпеливыми» заявками. В СМО с «нетерпеливыми» заявками заявка может «уйти» из системы, если время пребывания её в СМО превысит некоторое допустимое значение, которое в общем случае может быть случайным или характеризоваться некоторым распределением. «Терпеливая» заявка, попав в СМО, непременно дожждётся обслуживания.

Процесс продвижения заявки от входа к выходу СМО происходит в соответствии с некоторым законом управления процессами в СМО, который задаётся дисциплинами ожидания и обслуживания. Дисциплина ожидания определяет порядок приёма заявок в систему и размещения их в очереди, дисциплина обслуживания – порядок выбора заявок из очереди и назначения их на обслуживание.

Совокупность обслуженных и потерянных (полностью необслуженных либо недообслуженных) заявок образует выходящий поток СМО. В зависимости от структуры выходящего потока различают СМО без потерь («чистые» СМО) и СМО с потерями («смешанные» СМО).

Параметры и характеристики систем массового обслуживания

Параметры входящего потока. Процесс поступления заявок на обслуживание является в общем случае случайным и может рассматриваться как поток однородных событий, происходящих через случайные промежутки времени. Случайные временные интервалы между поступлениями заявок могут подчиняться различным законам распределения.

Наибольшее распространение в теории массового обслуживания получил простейший поток заявок, то есть поток, обладающий свойствами стационарности, ординарности и отсутствия последствия.

Если входящий поток представляет собой совокупность M потоков различных типов с интенсивностями λ_i , $i = \overline{1, M}$, то его можно характеризовать суммарной интенсивностью:

$$\lambda = \sum_{i=1}^M \lambda_i .$$

К параметрам входящего потока причислим также допустимое время пребывания $\tau_{\text{доп}}$ заявки в системе массового обслуживания, рассматривая его как свойство заявки. По истечении времени пребывания попавшая в систему заявка покидает её, не дожидаясь назначения на обслуживание или даже прерывая начатое обслуживание.

Параметры структуры СМО. Каждая система массового обслуживания обладает определённой структурой, характеризующейся совокупностью параметров. По составу СМО можно разделить на СМО с одним каналом обслуживания (одноканальные СМО) и с m каналами обслуживания (многоканальные СМО). В свою очередь, многоканальные СМО могут содержать одинаковые и различные по своей производительности каналы обслуживания.

Производительность канала обслуживания обратно пропорциональна длительности обслуживания. Эта случайная величина с функцией распределения $F(\tau)$, плотностью распределения $f(\tau)$ и математическим ожиданием $\tau_{об}$. Типы заявок различаются законами распределения. При этом принимается допущение о независимости длительности обслуживания для различных заявок одного типа, вполне корректное для большинства систем. Наряду с математическим ожиданием длительности обслуживания используется понятие интенсивности потока обслуживания – величина, характеризующая количество заявок, которое может быть обслужено в единицу времени постоянно загруженным каналом.

Важная компонента структуры СМО – очередь, параметром которой является число мест n . В приоритетных системах общая очередь может быть разделена на несколько очередей по числу различаемых приоритетов, для каждой из которых должно быть указано число мест n_i , $i = \overline{1, N}$. Ограничение на число мест может задаваться на всю совокупность очередей. При неограниченном числе мест в очереди (бесконечная очередь) отсутствуют потери заявок за счёт отказов и выталкиваний.

Закон управления процессами в СМО можно представить совокупностью двух дисциплин: дисциплины ожидания и дисциплины обслуживания. В беспriorитетных дисциплинах заявки какого-либо типа не имеют преимуществ перед заявками других типов при постановке в очередь или назначении на обслуживание. Если по каким-либо причинам заявки некото-

рых типов должны обслуживаться СМО быстрее, то они получают преимущество перед заявками других типов, называемое *приоритетом*. Приоритеты заявок характеризуются положительными числами 1, 2, 3, ..., причём более высокому приоритету соответствует меньшее число. Дисциплины ожидания и обслуживания, учитывающие приоритеты, называются приоритетными.

Дисциплина ожидания определяет правила управления очередью, возникающей в том случае, когда каналы обслуживания не справляются с потоком заявок.

Дисциплина обслуживания определяет правила выбора заявки из очереди при назначении на обслуживание. При бесприоритетной дисциплине обслуживания заявки различных типов не имеют привилегий перед заявками других типов на досрочное обслуживание. В приоритетных дисциплинах обслуживания заявкам некоторых типов предоставляется преимущественное право на обслуживание перед заявками других типов, называемое *приоритетом*. Различают относительные, абсолютные и смешанные приоритеты.

Относительные приоритеты учитываются только в момент назначения заявки на обслуживание. При освобождении канала обслуживания сравниваются приоритеты заявок, находящихся в очереди в состоянии ожидания, и обслуживание предоставляется заявке с наибольшим приоритетом.

Абсолютные приоритеты предполагают прерывание обслуживания низкоприоритетной заявки в момент поступления в СМО заявки с более высоким приоритетом, прерванная заявка ставится в начало либо общей очереди, либо очереди заявок соответствующего приоритета. Обслуживание прерванных заявок может производиться либо от начала (повторное обслуживание), либо от момента прерывания (дообслуживание).

Смешанные приоритеты предполагают сочетание рассмотренных видов приоритетов, причём для отдельных заявок может быть использовано бесприоритетное обслуживание.

Характеристики СМО могут быть разделены на две группы. Характеристики первой группы используются при рассмотрении СМО как элемента более сложной структуры, например сети массового обслуживания. Характеристики второй группы позволяют оценить способность конкретной СМО к выполнению возложенных на неё функций и называются *показателями эффективности*.

Важными характеристиками первой группы являются характеристики выходящего потока заявок. Выходящий поток заявок в общем случае распадается на поток обслуженных и поток потерянных заявок, каждый из которых характеризуется законом распределения длительности между соседними заявками. Если входящий поток содержит заявки M типов с интенсивностями λ_i потока заявок типов i , $i = \overline{1, M}$, то выходящий поток можно характеризо-

вать суммарной интенсивностью потока обслуженных заявок и суммарной интенсивностью потока потерянных заявок соответственно:

$$\lambda_{об} = \sum_{i=1}^M \lambda_{об}^i, \quad \lambda_{ни} = \sum_{i=1}^M \lambda_{ни}^i,$$

где $\lambda_{об}^i$ – интенсивность потока обслуженных заявок типа i ; $\lambda_{ни}^i$ – интенсивность потока потерянных заявок типа i .

Очевидно, что $\lambda_{об} + \lambda_{ни} = \lambda$.

Под *показателем эффективности* понимается количественный показатель, частично характеризующий уровень выполнения системой массового обслуживания возложенных на неё функций. На основании показателей эффективности можно построить некоторый критерий эффективности, совокупно характеризующий эффективность СМО при ограничениях на её параметры. Эффективность СМО можно характеризовать большим числом различных показателей эффективности. Рассмотрим наиболее употребительные из них.

Вероятность обслуживания $p_{об}$ – это вероятность того, что произвольно выбранная из входящего потока с интенсивностью λ заявка будет обслужена, то есть окажется в потоке обслуженных заявок с интенсивностью $\lambda_{об}$:

$$p_{об} = \frac{\lambda_{об}}{\lambda}.$$

Иногда вероятность обслуживания называют относительной пропускной способностью.

Вероятность потери p_n – это вероятность того, что произвольно выбранная из входящего потока с интенсивностью λ заявка окажется в потоке потерянных заявок с интенсивностью $\lambda_{ни}$:

$$p_n = \frac{\lambda_{ни}}{\lambda} = \frac{\lambda - \lambda_{об}}{\lambda} = 1 - \frac{\lambda_{об}}{\lambda} = 1 - p_{об}.$$

Среднее время ожидания $\overline{t_{ож}}$ заявки (среднее время пребывания заявки в очереди) – математическое ожидание времени ожидания. Время ожидания $t_{ож}$ заявки – случайная величина, равная сумме длительностей интервалов времени, в течение которых заявка находится в очереди, начиная с момента появления заявки на входе СМО и кончая моментом, когда она последний раз покидает очередь по причине назначения на обслуживание, ухода из очереди (в случае нетерпеливых заявок) либо выталкивания низкоприоритетной заявки высокоприоритетной заявкой для некоторых приоритетных дисциплин ожидания.

Среднее время пребывания заявки в СМО $\overline{t_c}$ – математическое ожидание времени пребывания заявки в СМО, равное промежутку времени от момента поступления заявки на

вход СМО до момента появления её в выходном потоке и связанное с длительностью процессов ожидания $t_{ож}$ и обслуживания $t_{об}$. Среднее время пребывания заявки в СМО равно сумме среднего времени ожидания (пребывания в очереди) и среднего времени обслуживания (пребывания в канале обслуживания):

$$\overline{t_c} = \overline{t_{ож}} + \overline{t_{об}} .$$

Критерием эффективности СМО является некоторая функция показателей эффективности, которая служит для совокупной оценки пригодности СМО к выполнению возложенных на неё функций. Выбор того или иного критерия эффективности зависит от назначения СМО, условий её функционирования и т.д. Рассмотрим примеры критериев эффективности, применимых к вычислительным системам, работающим в режиме реального времени.

Пусть заявка обесценивается пропорционально её задержке в СМО, то есть время пребывания заявки в СМО (иногда учитывают лишь время ожидания в очереди). Тогда эффективность E СМО равна:

$$E = \sum_{i=1}^M e_{ci} \overline{t_{ci}} ,$$

где e_{ci} – штраф за единицу времени пребывания заявки i -го типа в СМО; $\overline{t_{ci}}$ – среднее время пребывания в СМО заявок. Штрафу может подвергаться потеря заявки системой, иногда отдельно штрафуют потери за счёт отказов, выталкиваний и уходов. Возможен функционал вида:

$$E' = \sum_{i=1}^M \left[e_{omki} p_{отки} \lambda_i + e_{vi} p_{vi} \lambda + e_{yi} p_{yi} \lambda_i \right] ,$$

где e_{omki} – штраф за отказ СМО принять заявку i -го типа; e_{vi} – штраф за выталкивание из очереди заявки; e_{yi} – штраф за уход из СМО нетерпеливой заявки; $p_{отки}$, p_{vi} , p_{yi} – вероятности соответственно отказа, выталкивания и ухода заявки i -го типа; λ_i – интенсивность входящего потока заявок i -го типа.

Критерии эффективности рассмотренных типов могут, в свою очередь, объединяться в новые, более сложные критерии эффективности.

Как характеристики, так и критерии эффективности СМО могут рассматриваться в одном из двух режимов работы СМО. Первый из режимов, называемый *переходным*, соответствует начальному этапу функционирования СМО. Начальными условиями для него являются отсутствие очереди и свободные от обслуживания каналы обслуживания. Режим, следующий за переходным, называется *установившимся* (следует отметить, что установив-

шийся режим в СМО возможен не всегда); поведение системы в этом режиме характеризуется обычно средними значениями характеристик, отражающими устойчивые свойства СМО.

Программа для моделирования системы «Человек – машина»

Программа предназначена для моделирования СМО с одним или несколькими каналами обслуживания и до десяти типов заявок. Причём можно задавать параметры заявок каждого типа.

Для запуска программы войдите в среду *Windows* и выполните файл *SMO32.EXE*. После запуска программы перед вами появится окно с тремя закладками: «Установки», «Эффективность» и «Статистика».

В закладке «Установки» введите параметры СМО: интенсивности заявок, их приоритеты, время обслуживания, размер очереди. Можно сохранить установки в файле или загрузить их из ранее записанного файла. Для этого выберите из меню «Файл» соответствующий пункт.

После ввода характеристик СМО можно приступить к моделированию. Для этого выберите из меню пункт «Моделирование | Моделирование». Для приостановки моделирования выберите пункт меню «Моделирование | Пауза» (для продолжения выберите этот пункт ещё раз), а для остановки – пункт «Моделирование | Остановка моделирования».

В закладке «Статистика» показывается вся статистика процесса моделирования: количество сгенерированных и обработанных заявок, заполнение очереди, время ожидания заявок в очереди и количество заявок, покинувших СМО без обслуживания.

Здесь колонки для каждого типа заявок означают:

- поступило заявок – показывает количество заявок данного типа, поступивших на данный момент в систему. В строке «Итого/среднее» – общее количество заявок, поступивших в систему на данный момент времени;

- обработано заявок – показывает количество заявок данного типа, обработанных системой в данный момент. В строке «Итого/среднее» – общее количество заявок, обработанных системой на данный момент времени;

- покинуло без обслуживания – показывает количество заявок данного типа, которые покинули систему без обслуживания на данный момент. В строке «Итого/среднее» – общее количество заявок, покинувших систему без обслуживания на данный момент времени;

- заполнение очереди, % – показывает, на сколько процентов заполняют общую очередь заявки данного типа на данный момент времени. В строке «Итого/среднее» – процент заполнения очереди всеми заявками на данный момент времени;

- среднее время ожидания, c – сколько времени в среднем заявка данного типа провела в очереди, дожидаясь обслуживания на данный момент времени. В строке «Итого/среднее» – среднее время пребывания заявки в очереди СМО на данный момент времени;

- загрузка каналов обслуживания, % – показывает, сколько процентов времени каналы обслуживания занимают обработкой заявок. В строке «Итого/среднее» – общий процент загрузки каналов СМО по всем заявкам на данный момент времени;

- канал обслуживания – если заявка данного типа в данный момент времени обслуживается, то здесь указывается номер канала обслуживания, который обслуживает её.

В закладке «Эффективность» вам необходимо ввести показатели для вычисления критерия эффективности системы. Это штрафы за ожидание заявок в очередях и простой каналов обслуживания. Критерий эффективности вычисляется по формуле:

$$E' = \sum_{i=1}^M \left[p_{откi} \lambda_i + e_{ci} p_{vi} \lambda_i + e_{yi} p_{yi} \lambda_i \right],$$

где e_{ci} – штраф за единицу времени пребывания заявки i -го типа в СМО; \bar{t}_{ci} – среднее время пребывания в СМО заявок; A – штраф за 1% недогрузки каналов обслуживания; \bar{K} – средняя загрузка всех каналов обслуживания, %.

Можно сохранить статистику моделирования после остановки моделирования или во время паузы. Для этого выберите пункт меню «Файл | Сохранение статистики». Для просмотра статистики надо воспользоваться «Блокнотом» *Windows* или любым другим редактором под *Windows* (например, *Word*).

Для выхода из программы выберите пункт «Файл | Выход».

Пример решения

В программе выбрать «Файл» → «Открытие файла установок...», выбрать файл *example.SMO*.

Содержимое файла:
 [Версия]
 Версия=2
 Подверсия=3
 [Общие]
 Генераторы=2
 Каналы=2
 Время=1
 ВремяЧ=1000
 ВремяМ=0
 ВремяС=0
 Очередь=1000
 [Заявки]
 Имя1=Генератор N 1
 Период1=6000

Разброс1=700
 Приоритет1=10
 ОбработкаЧ1=7500
 ОбработкаМ1=2500
 Имя2=Генератор N 2
 Период2=500000
 Разброс2=55000
 Приоритет2=4097
 ОбработкаЧ2=100000
 ОбработкаМ2=100000
 [Критерий эффективности]
 Недогрузка=100
 Ожидание1=1
 Ожидание2=500
 [Другое]
 Обновление=1
 Показ каналов=1

Нажать на кнопку «Начало моделирования» (или пункт меню «Моделирование» → «Моделирование»). Дождаться конца моделирования.

На вкладке «Эффективность» можно увидеть значение критерия эффективности: около 49500. На вкладке «Статистика» можно увидеть, что загрузка каналов была максимальной, в то время как среднее время ожидания заявок первого типа составляет как раз около 49500. Следовательно, на критерий эффективности повлияло ожидание в очереди заявок первого типа.

Уменьшить значение критерия эффективности, изменяя дисциплину обслуживания заявок: изменяя приоритеты.

В текущий момент заявки второго типа имеют абсолютный приоритет (над заявками первого типа), что означает, что поступление заявки второго типа прерывает обработку заявки первого типа и блокирует канал обслуживания за собой.

Будем изменять приоритеты заявок. Результаты представлены в таблице.

Генератор 1		Генератор 2		Значение критерия эффективности
отн. / абс.	значение	отн. / абс.	значение	
отн	2	абс	1	49500
отн	1	отн	1	22 млн.
отн	1	отн	2	152 млн.
отн	2	отн	1	61000
абс	1	абс	1	22 млн
абс	1	абс	2	15 млрд.
абс	2	абс	1	49500

По результатам моделирования видно, что текущая дисциплина обслуживания заявок является оптимальной.

Уменьшить значение критерия эффективности изменяя количества каналов обслуживания. Текущее значение – 2.

Результаты моделирования приведены в таблице.

Число каналов обслуживания	Значение критерия эффективности
1	109000
2	49500
3	3110
4	4835
5	5665

По результатам моделирования видно, что добавление ещё одного канала обслуживания увеличит недогрузку каналов с 0% до 31%, однако снижает значение критерия эффективности до 3110.

3. Порядок выполнения работы

- 1) Ознакомиться с программой для моделирования системы «Человек – машина».
- 2) Получить у преподавателя вариант задания.
- 3) Открыть файл с заданием (вариантам с 1 по 10 соответствуют файлы с 1.SMO по 10.SMO) и загрузить свой вариант задания.
- 4) Выполнить моделирование.
- 5) Сохранить результаты моделирования в файле.
- 6) Проанализировать зависимость характеристик СМО от изменения входных показателей (законов обслуживания, типов заявок и количества каналов обслуживания). Результаты отобразить графически.
- 7) Найти оптимальное (минимальное) значение критерия эффективности (для выданного задания) при различном количестве каналов обслуживания.

- 8) Оформить отчёт по выполненной работе

4. Содержание отчёта по лабораторной работе

- 1) Название, цель работы, номер варианта, задание.
- 2) Назначение систем моделирования.
- 3) Формализация задачи в терминах теории массового обслуживания.
- 4) Графики зависимости критерия эффективности системы. Анализ результатов моделирования
- 5) Выводы по результатам исследований.

5. Контрольные вопросы

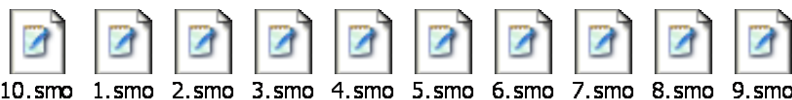
- 1) Опишите общие сведения о теории массового обслуживания.
- 2) Какие параметры входящего потока Вы знаете?
- 3) Какие параметры структуры СМО Вы знаете?
- 4) Охарактеризуйте параметры закона управления процессами в СМО.
- 5) Назовите характеристики СМО.
- 6) Какие показатели эффективности СМО Вы знаете?
- 7) Какие критерии эффективности СМО Вы знаете?

6. Список литературы

1. Мамиконов А.Г. Основы построения АСУ. – М.: Высш. шк., 1981. – 248 с.
2. Организация взаимодействия человека с техническими средствами АСУ / Под ред. В.Н. Четверякова. – М.: Высш. шк., 1990. – 127 с.
3. Вентцель Е.С. Теория вероятностей. – М.: Наука, 1969. – 576 с.
4. Вероятностные методы в вычислительной технике / Под ред. А.Н. Лебедева, Е.А. Чернявского. – М.: Высш. шк., 19с.

7. Варианты индивидуальных заданий

Варианты индивидуальных заданий приведены в файлах 1.smo – 10.smo



Лабораторная работа №2

Построение нечетких моделей

1. Цель работы

Ознакомление магистрантов с методами разработки математических моделей на основе использования математического аппарата нечетких систем в программе *MATLAB* 7.0.1

2. Теоретические сведения

В *MATLAB* важная роль отводится специализированным группам программ, называемых *toolboxes*. *Toolboxes* – это всесторонняя коллекция функций *MATLAB* (*M*-файлов), которые позволяют решать частные классы задач [2].

Пакет прикладных программ *Fuzzy Logic* относится к теории нечетких (размытых) множеств. Обеспечивается поддержка современных методов нечеткой кластеризации и адаптивных нечетких нейронных сетей. Графические средства пакета позволяют интерактивно отслеживать особенности поведения системы.

Основные возможности пакета:

- 1) определение переменных, нечетких правил и функций принадлежности;
- 2) интерактивный просмотр нечеткого логического вывода;
- 3) современные методы: адаптивный нечеткий вывод с использованием нейронных сетей, нечеткая кластеризация;
- 4) интерактивное динамическое моделирование в *Simulink*.

Создание *m*-функции для моделирования. *M*-файлы являются обычными текстовыми файлами, которые создаются с помощью текстового редактора. Для операционной среды персонального компьютера система *MATLAB* поддерживает специальный встроенный редактор/отладчик, хотя можно использовать и любой другой текстовый редактор с *ASCII*-кодами.

M-функции являются *M*-файлами, которые допускают наличие входных и выходных аргументов. Они работают с переменными в пределах *собственной рабочей области*, отличной от рабочей области системы *MATLAB*.

Построение нечётких моделей слабо формализовано. В каждом случае применяются свои параметры построения модели. В методе построения модели с использованием *FCM*-кластеризации можно управлять размерами выборки (количество наблюдений в выборке, количество входных переменных), используемой для построения модели, а также видом модели (Сугэно / Мамдани).

Одним из недостатков нечётких моделей является эффект «переобучения», когда модель даёт минимальную ошибку на элементах обучающего множества при большой ошибке на элементах тестирующего множества. Это означает, что модель «запомнила» обучающие

наборы вместо того, чтобы научиться решать поставленную задачу. Для преодоления этого недостатка применяют разбиение исходной выборки на два подмножества: обучающую и проверочную. Обучающая используется для построения и настройки модели, проверочная – для определения пригодности модели. Большие проблемы могут возникнуть при использовании большого числа входных переменных. При этом образуется огромное число внутренних элементов модели, сама модель становится относительно более медленной, затрудняется или становится вообще невозможным (при имеющихся технических мощностях) обучение модели.

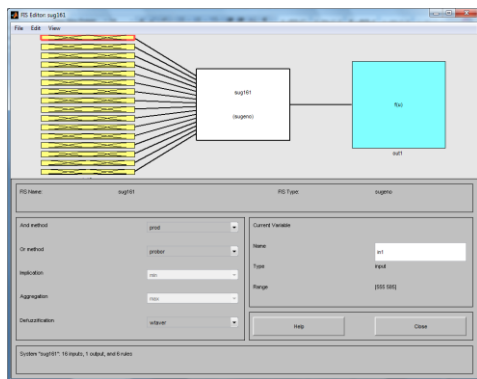
Пакет *MATLAB Fuzzy Logic Toolbox* позволяет строить модели тремя различными способами:

- 1) *genfis1* – построение нечёткой модели типа Сугэно, которую далее необходимо обучить с использованием *anfis*-обучения;
- 2) *genfis2* – генерация нечёткой модели с использованием алгоритма субтрактивной кластеризации;
- 3) *genfis3* – генерация нечёткой модели с использованием нечёткой (*FCM*) кластеризации.

Выбор способа построения нечеткой модели, определение параметров (отбор информативных входных переменных, размер обучающей выборки, величина радиуса центра кластеризации) для построения качественной модели должен проводиться путём постановки вычислительного эксперимента со статистическими данными, используемыми для построения модели [3, 4]. В качестве оценки адекватности модели можно использовать значение коэффициента парной корреляции, который будет показывать, насколько достоверно нечёткая модель моделирует выходную переменную.

При помощи инструмента *Fuzzy Logic* можно посмотреть сформированную структуру модели и нечеткие правила.

На рисунке 1 видна структура модели. На вход модели поступают 16 значений x_i , на выходе одно значение y .



Можно посмотреть сформированные нечеткие переменные и функции принадлежности (Рисунок 2). При этом заданы минимальные и максимальные значения для x_i , названия функций и переменных. Все функции принадлежности по умолчанию являются симметричными гауссовскими кривыми (*gaussmf*).

Рисунок 1 – Структура нечеткой модели

Также можно просмотреть сформированный набор правил (Рисунок 3). Видно, что все правила представляются видом:

Если (Выражение1) *И* (Выражение2)...*И* (Выражение16) *то* (Значение 1)(Значение 2)...(Значение m).

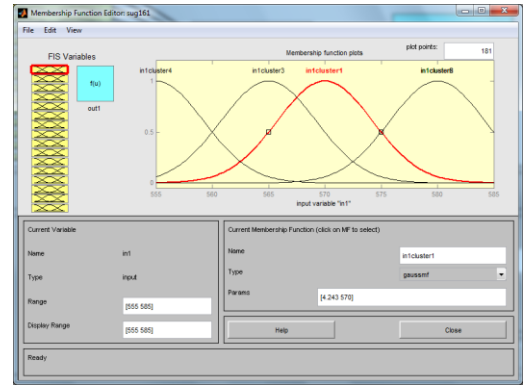


Рисунок 2 – Функции принадлежности нечеткой переменной x_1

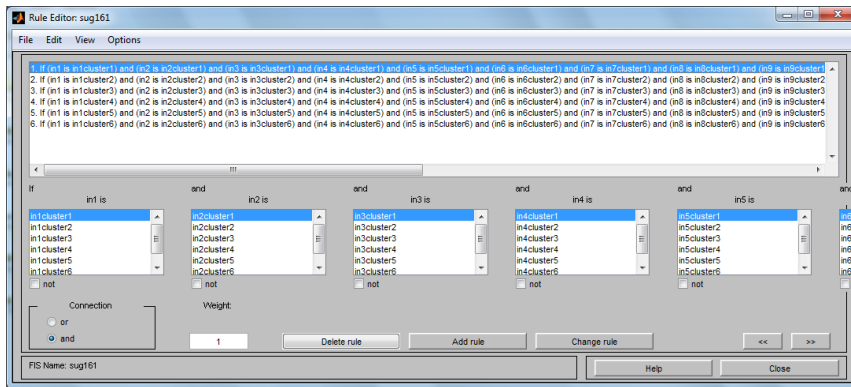


Рисунок 3 – Правила построения нечеткой модели

Пример. Имеются статистические данные функционирования многомерного объекта управления с семью входными переменными $X=(x_1, x_2, ..x_7)$ и с тремя выходными переменными $Y=(y_1, y_2, y_3)$ (таблица 1).

Таблица 1- Статистические данные объекта управления (обучающая выборка)

№п/п	x_1	x_2	x_3	x_4	x_5	x_6	x_7	y_1	y_2	y_3
1	1	655	0,1	1,22	0,3	0,35	5	245	89	40
2	1	679	0,2	1,2	0,35	0,35	7	253	91	35
3	1	644	0,21	1,22	0,3	0,35	7	246	83	38
4	1	644	0,17	1,11	0,2	0,4	7	230	81	39
5	1	635	0,2	1,21	0,2	0,4	6	245	78	37
6	1	655	0,2	1,21	0,2	0,4	7	215	75	42
7	1	655	0,2	1,21	0,2	0,4	7	210	70	45
8	1	655	0,2	1,21	0,2	0,4	7	208	68	51
9	1	650	0,22	1,24	0,3	0,35	15	189	61	55

10	1	672	0,22	1,2	0,3	0,35	16	150	59	57
11	1	679	0,22	1,2	0,3	0,35	15	198	62	52
12	1	635	0,22	1,2	0,3	0,35	15	157	60	53
13	1	644	0,21	1,22	0,3	0,35	15	190	71	48
14	1	644	0,17	1,11	0,2	0,4	15	150	59	59
15	1	575	0,17	1,11	0,2	0,4	15	139	55	60
16	1	635	0,18	1,11	0,2	0,4	15	184	67	54
17	1	635	0,2	1,21	0,2	0,4	14	193	68	49
18	1	655	0,2	1,21	0,2	0,4	14	200	71	37
19	2	650	0,2	1,21	0,3	0,35	17	145	59	46
20	2	650	0,2	1,21	0,3	0,35	17	137	55	53
21	2	650	0,2	1,21	0,3	0,35	17	124	51	58
22	2	650	0,2	1,21	0,3	0,35	18	132	55	64
23	2	650	0,2	1,21	0,4	0,4	18	126	54	62
24	2	650	0,2	1,21	0,4	0,4	18	117	51	67
25	2	650	0,2	1,21	0,4	0,4	18	113	49	65
26	2	650	0,2	1,21	0,4	0,4	18	119	52	66
27	2	650	0,2	1,21	0,4	0,4	18	127	61	67
28	2	650	0,2	1,21	0,4	0,4	18	118	53	64
29	2	650	0,2	1,21	0,4	0,4	18	125	51	68
30	2	650	0,2	1,21	0,4	0,4	18	131	53	69
31	2	650	0,2	1,21	0,4	0,4	18	120	54	65
32	2	650	0,2	1,21	0,4	0,4	18	121	49	69

Необходимо построить модель на нечетких множествах, описывающую зависимость выходных переменных y_1, y_2, y_3 от входных переменных x_1, x_2, \dots, x_7 . Проверить адекватность построенной модели и точность ее прогнозирования на проверочной выборке (таблица 2).

Таблица 2- Статистические данные объекта управления (проверочная выборка)

№п/п	x_1	x_2	x_3	x_4	x_5	x_6	x_7	y_1	y_2	y_3
1	1	655	0,1	1,22	0,3	0,35	5	245	89	40
2	1	679	0,2	1,2	0,35	0,35	7	253	91	35
3	1	644	0,21	1,22	0,3	0,35	7	246	83	38
4	1	644	0,17	1,11	0,2	0,4	7	230	81	39
5	1	635	0,2	1,21	0,2	0,4	6	245	78	37

6	1	655	0,2	1,21	0,2	0,4	7	215	75	42
7	1	655	0,2	1,21	0,2	0,4	7	210	70	45
8	1	655	0,2	1,21	0,2	0,4	7	208	68	51
9	1	650	0,22	1,24	0,3	0,35	15	189	61	55
10	1	672	0,22	1,2	0,3	0,35	16	150	59	57

1) Вызвать программу *matlab*, создать новый файл с названием *primer* разместив в нем два файла: с входными переменными под названием *datin* и файл с выходными переменными y_1, y_2, y_3 - под названием *datout*. Исходные данные копировать с таблицы 1. Для создания нового файла с данными, в меню *matlab* вызывать *File\New\Variable*. При этом появится окно с пустым файлом *unnamed*. Вызвать контекстное меню командой *Rename* и переименовать файл *datin*. Нажав дважды мышкой раскрыть пустой файл. В появившуюся таблицу записать входные данные, скопированные с таблицы 1.

Просмотреть графики входных переменных с целью визуальной оценки наличия «грубых» данных: выбрать столбец с анализируемыми данными, вызвать контекстное меню, выполнить команду *plot(datin(1:32,n))*, где $n=1,2,..7$ номер столбца с входными данными. При этом на экране появится график выбранной входной переменной. При обнаружении грубых данных их скорректировать путем замены на среднее арифметическое, вычисленное по «соседним» данным. Затем закрыть таблицу с данными.

Нажать иконку с названием *New Variable* и создать второй файл с названием *datout*, заполнив его выходными данными y_1, y_2, y_3 , скопированными с таблицы 1. Просмотреть график выходной переменной с целью оценки наличия грубых данных и их корректировки при необходимости. Сохранить оба созданных файла в папке с отчетом о выполняемой лабораторной работе под общим названием *primer* с помощью команды *matlab: File\Save\Workspase As....*

2) Создать модель на нечетких множествах. Для создания модели использовать функцию *genfis2*. Функция *genfis2* генерирует систему нечеткого логического вывода типа «Сугэно» из данных с использованием субтрактивной кластеризации. Экстракция правил из данных в функции *genfis2* происходит в два этапа. Вначале используется функция *subclust* для определения количества правил и мощностей терм-множеств выходных переменных. Затем с помощью метода наименьших квадратов определяется "то-" часть каждого правила. В результате этого получается система нечеткого логического вывода с базой правил, покрывающих всю предметную область.

Функция *genfis2* может иметь до пяти входных аргументов, первые три из которых обязательны:

- X_{in} – матрица, в которой каждая строка содержит значения входных переменных;
- X_{out} – матрица, в которой каждая строка содержит значения выходных переменных;

- $radii$ – вектор, определяющий размеры области правил по каждой координате. Значения координат вектора $radii$ должны находиться в диапазоне $[0, 1]$ в связи с тем, что во время выполнения функции $subclust$ данные масштабируются на единичный гиперкуб. Если значение $radii$ задано скаляром, тогда все координаты считаются равноважными;

- $xBounds$ – матрица диапазонов изменения данных, необходимая для их масштабирования на единичный гиперкуб. Каждый столбец матрицы задает диапазон изменения данных по одной координате, т.о. размер матрицы – $2 \times p$, где p – количество входных и выходных переменных. Если аргумент $xBounds$ не задан, тогда диапазоны изменения данных рассчитываются функцией $subclust$ по фактическим значениям матриц X_{in} и X_{out} [1].

M -файлы являются обычными текстовыми файлами, которые создаются с помощью текстового редактора.

Предварительно определить диапазон изменения переменных, используемых для построения модели. Для чего в командном окне *Command Window* набрать команды *File\New\M-File*. В текстовом редакторе *Editor – Untitled* набрать команды:

```
i=min(datin)
j=max(datin)
a=[i:j]
```

При этом в файл *matlab primer* занесутся файлы с названиями i , j , a . Открыть файл a , скопировать данные и записать их в таблицу с диапазоном изменения входных переменных (таблица3).

Таблица 3- Диапазон изменения входных переменных $datin$

Переменные	x_1	x_2	x_3	x_4	x_5	x_6	x_7
Минимальное значение	1	575	0,1	1,11	0,2	0,35	5
Максимальное значение	2	679	0,22	1,24	0,4	0,4	18

Аналогично определить диапазон изменения выходных переменных (таблица 4).

Таблица 4 - Диапазон изменения выходных переменных *datout*

Переменные	y1	y2	y3
Минимальное значение	113	49	35
Максимальное значение	253	91	69

Построить модель на нечетких множествах в *M-File model*:

```
expert =genfis2(datin,datout,0.44,[1 575 0.1 1.11 0.2 0.35 5 113 49 35; 2 679 0.22 1.24 0.4 0.4  
18 253 91 69])
```

3) Просмотреть структуру модели набрав в командном окне:

Fuzzy (expert)

При этом на экране *Fis Editor:sug 73* отобразится структура модели, аналогичная рисунку 3.1, с семью входными и тремя выходными переменными.

Просмотреть сформированные нечеткие переменные и функции принадлежности, набрав команду:

Edit\Membership Functions

При этом на экране отобразятся функции принадлежности, аналогичные рисунку 2, для каждой входной переменной, выбираемой клавишей *FIS Variables*.

Ознакомиться с сформированным набором правил вывода, набрав команду:

Edit\Rules

При этом на экране отобразятся правила построения нечеткой модели, аналогичные рисунку 3.

4) Рассчитать выходные значения переменных *y1*, *y2*, *y3*, записав их в файл *modout*, на полученной модели *expert*:

```
modout = evalfis(datin, expert)
```

Функция *evalfis* выполняет нечеткий логический вывод, может иметь три входных аргумента, два из которых обязательные:

- *datin* – матрица значений входных переменных, для которых необходимо выполнить нечеткий логический вывод. Матрица должна иметь размер $m \times n$, где n – количество входных переменных; m – количество входных данных. Каждая строчка матрицы представляет один вектор значений входных переменных;

- *expert* – идентификатор системы нечеткого логического вывода.

Оценить качество разработанной модели, для чего построить совмещенные графики выходных переменных по фактическим *datout* и рассчитанным по модели *expert* данным *modout*:

```
subplot(2,1,1);
plot(datout);
legend('show');
```

```
subplot(2,1,2);
plot(modout);
legend('show');
```

Файлы с выходными данными *datout* и *modout* скопировать в *EXCEL* и рассчитать коэффициенты детерминации по выходным переменным по формуле:

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - y_{i,mod})^2}{\sum_{i=1}^n (y_i - y_{cp})^2}, \quad (1)$$

где y_i – фактическое значение выходной переменной в обучающей выборке, $i=1, 2, 3$; $y_{i,mod}$ – значение переменной на выходе модели, $i=1, 2, 3$; y_{cp} – среднее арифметическое значение выходной переменной в обучающей выборке.

Проверка значимости коэффициента детерминации осуществляется путем сравнения расчетного значения критерия Фишера:

$$F_{расч} = \frac{R^2 / n - m}{1 - R^2 / m - 1}, \quad (2)$$

с табличным $F_{табл}$. Табличное значение критерия определяется заданным уровнем значимости $\alpha = 0,05$ и степенями свободы $k_1 = m - 1$ и $k_2 = n - m$, где n – количество данных в обучающей выборке, $m = 7$ – количество входных переменных. Коэффициент R^2 значительно отличается от нуля, если выполняется неравенство

$$F_{расч} > F_{табл}.$$

Коэффициент детерминации показывает долю вариации выходной переменной, находящегося под воздействием входных переменных, т. е. определяет, какая доля вариации Y учтена в модели и обусловлена влиянием на него входных переменных. Чем ближе R^2 к 1, тем выше качество модели. Данный коэффициент является универсальным, так как он отражает тесноту связи и точность модели, а также может использоваться при любой форме связи переменных.

Коэффициент детерминации модели, рассчитанный по выходной переменной y_1 , равен $R^2=0,95$, статистически значим с уровнем значимости 0,05. Большое значение коэффициента множественной корреляции свидетельствует о том, что вариация переменной y_1 в обучающей выборке на 95% определяется вариацией входных переменных $x_1- x_7$. Влияние неучтенных факторов незначительное, составляет 5%.

5) Определить пригодность разработанной модели на проверочной выборке, приведенной в таблице 3.2, предварительно создав файлы для входных переменных *chkdatin* и выходных переменных *chkdatout*. Рассчитать (прогнозировать) значения выходных переменных и записать их в файл *chkfuzout*:

chkfuzout=evalfis(chkdatin, expert);

Построить совмещенные графики выходных переменных по фактическим *chkdatout* и рассчитанным (прогнозированным) по модели *expert* данным *chkfuzout*:

subplot(2,1,1);
plot(chkdatout);
legend('show');

subplot(2,1,2);
plot(chkfuzout);
legend('show');

Файлы с выходными проверочными *chkdatout* и прогнозными *chkfuzout* данными скопировать в EXCEL. Рассчитать дисперсию погрешности модели s^2 и коэффициент парной корреляции r по выходным переменным y_1, y_2, y_3 по формулам:

$$s^2 = \frac{\sum_{i=1}^n (y_i - y_{i, \text{мод}})^2}{n}, \quad (3)$$

$$r = \frac{\sum_{i=1}^n (y_i - y_{cp}) (y_{i, \text{мод}} - y_{\text{мод, ср}})}{S S_{\text{мод}}}, \quad (4)$$

где y_i – значения выходной переменной в проверочной выборке, $i=1, 2, 3$; $y_{i \text{ мод}}$ – прогнозируемое значение переменной на выходе модели, $i=1, 2, 3$; y_{cp} – среднее арифметическое значение выходной переменной в проверочной выборке; $y_{\text{мод, ср}}$ – среднее арифметическое значение прогнозной переменной на выходе модели; s – стандартное отклонение переменных в проверочной выборке; $s_{\text{мод}}$ – стандартное отклонение прогнозных переменных; n – количество данных в проверочной выборке.

Значимость коэффициента парной корреляции проверяется по t -критерию Стьюдента:

$$t_{\text{набл}} = \sqrt{\frac{r_{y,x}^2}{1-r_{y,x}^2}}(n-2) \quad (5)$$

Вычисленное по этой формуле значение $t_{\text{набл}}$ сравнивается с критическим значением t -критерия, которое берется из таблицы значений t Стьюдента с учетом заданного уровня значимости 0,05 и числа степеней свободы $(n-2)$.

Если $t_{\text{набл}} > t_{\text{кр}}$, то полученное значение коэффициента корреляции признается значимым (то есть нулевая гипотеза, утверждающая равенство нулю коэффициента корреляции, отвергается). И таким образом делается вывод о том, что между исследуемыми переменными есть тесная статистическая взаимосвязь.

Дисперсия погрешности модели, рассчитанная по выходной переменной y_1 , равна $s^2 = 150,5$. Коэффициент парной корреляции между y_1 и y_1 мод равен $r = 0,92$, статистически значим с уровнем значимости 0,05. Большое значение коэффициента парной корреляции свидетельствует о том, что модель с высокой точностью описывает характер изменения выходной переменной y_1 на проверочной выборке. Это подтверждает высокую точность разработанной модели на нечетких множествах.

3. Задание к лабораторной работе

Разработать математическую модель на основе использования математического аппарата нечетких систем в программе *MATLAB* 7.0.1. Оценить адекватность разработанной модели на обучающей выборке. Оценить точность модели на проверочной выборке. Составить отчет по лабораторной работе и сделать выводы по результатам исследований.

4. Порядок выполнения лабораторной работы

Получить от преподавателя индивидуальное задание. Ознакомиться с методом построения нечеткой модели в программе *MATLAB* 7.0.1.

1) Создать в *MATLAB* файл с исходными данными. Проверить данные на наличие «грубых» ошибок.

2) Определить диапазон изменения переменных. Построить многомерную нечеткую модель, описывающую зависимость выходных переменных от множества входных переменных. Записать модель в m -файл.

3) Проанализировать структуру разработанной нечеткой модели, функции принадлежности и сформированный набор правил вывода.

4) Рассчитать выходные значения переменных y_1, y_2, y_3 на созданной нечеткой модели. Оценить качество разработанной модели по заданной выходной переменной.

5) Определить пригодность разработанной модели на проверочной выборке по заданной выходной переменной по величине дисперсии погрешности модели и коэффициенту парной корреляции между проверочными и модельными данными.

6) Составить отчет по лабораторной работе.

5. Содержание отчета

1) Индивидуальное задание на лабораторную работу.

2) Таблица с анализом исходных данных, используемых для построения модели на нечетких множествах.

3) Текстовый m - файл с m -функцией для моделирования многомерного объекта управления.

4) Структура разработанной нечеткой модели, функции принадлежности и сформированный набор правил вывода.

5) Расчеты значения коэффициента детерминации модели по заданной выходной переменной и оценка его значимости при уровне значимости 0,05.

6) Расчеты точности модели на проверочной выборке, оценка тесноты стохастической связи прогнозных данных с проверочными данными.

7) Выводы по результатам исследований разработанной модели объекта управления на нечетких множествах.

6. Вопросы для самоконтроля

1) В каких случаях используются нечеткие системы для разработки моделей объектов управления.

2) Что используется в качестве входных и выходных сигналов при разработке модели объекта управления.

3) Структура нечеткой системы с фуззификатором и дефуззификатором.

4) Фуззификатор, его функция в структуре нечеткой системы.

5) Назначение функции принадлежности в нечеткой системе и их виды.

6) Назначение дефуззификатора в структуре нечеткой системы.

7) Функция *genfis2* пакета *MATLAB Fuzzy Logic Toolbox*, используемая для построения модели. Входные аргументы функции.

8) Назначение функция *evalfis*. Обязательные аргументы функции.

9) Назначение обучающей и проверочной выборок при построении модели в нечеткой системе.

10) Оценка адекватности и точности модели объекта управления, созданной в нечеткой системе.

7. Список рекомендуемой литературы

- 1) Осовский С. Нейронные сети для обработки информации /Пер. с польского И.Д. Рудинского. -М.: Финансы и статистика, 2002. -344с.
- 2) Перевод Конюшенко В.В. Начало работы с MATLAB. *Online* Документация по программному продукту. [Электронный ресурс]. – URL: <http://matlab.exponenta.ru/ml/book3/index.php>
- 3) Чушикина А.Ю. Исследование и разработка системы поддержки принятия решений на основе нечеткого обратного вывода /магистерская диссертация. Владим. гос. ун-т, Владимир, 2012.
- 4) Фоминой Н. А. Исследование эффективности применения моделей на нечетких множествах при управлении технологическими системами / магистерская диссертация. Владим. гос. ун-т, Владимир, 2011.

8. Варианты заданий к лабораторной работе

Задание 1. Построить нечеткую модель объекта управления с семью входными x_1 - x_7 и тремя выходными y_1 - y_3 переменными на основе использования математического аппарата нечетких систем в программе *MATLAB* 7.0.1. Оценить точность разработанной модели по выходной переменной y_1 . Исходные данные для построения модели и проверки точности приведены соответственно в таблицах 3 и 4.

Таблица 3- Статистические данные объекта управления (обучающая выборка)

№ п/п	x_1	x_2	x_3	x_4	x_5	x_6	x_7	y_1	y_2	y_3
1	1	655	0,1	1,22	0,3	0,35	5	245	89	40
2	1	679	0,2	1,2	0,35	0,35	7	253	91	35
3	1	644	0,21	1,22	0,3	0,35	7	246	83	38
4	1	635	0,2	1,21	0,2	0,4	6	245	78	37
5	1	650	0,22	1,24	0,3	0,35	15	189	61	55
6	1	679	0,22	1,2	0,3	0,35	15	198	62	52
7	1	635	0,22	1,2	0,3	0,35	15	157	60	53
8	1	644	0,21	1,22	0,3	0,35	15	190	71	48
9	1	575	0,17	1,11	0,2	0,4	15	139	55	60
10	1	635	0,18	1,11	0,2	0,4	15	184	67	54

11	1	635	0,2	1,21	0,2	0,4	14	193	68	49
12	1	655	0,2	1,21	0,2	0,4	14	200	71	37
13	2	650	0,2	1,21	0,3	0,35	17	137	55	53
14	2	650	0,2	1,21	0,3	0,35	17	124	51	58
15	2	650	0,2	1,21	0,4	0,4	18	126	54	62
16	2	650	0,2	1,21	0,4	0,4	18	117	51	67
17	2	650	0,2	1,21	0,4	0,4	18	113	49	65
18	2	650	0,2	1,21	0,4	0,4	18	118	53	64
19	2	650	0,2	1,21	0,4	0,4	18	125	51	68
20	2	650	0,2	1,21	0,4	0,4	18	131	53	69
21	2	650	0,2	1,21	0,4	0,4	18	120	54	65
22	2	650	0,2	1,21	0,4	0,4	18	121	49	69

Таблица 4- Статистические данные объекта управления (проверочная выборка)

№ п/п	x_1	x_2	x_3	x_4	x_5	x_6	x_7	y_1	y_2	y_3
1	1	644	0,17	1,11	0,2	0,4	15	150	59	59
2	2	650	0,2	1,21	0,3	0,35	17	145	59	46
3	2	650	0,2	1,21	0,3	0,35	18	132	55	64
4	1	644	0,17	1,11	0,2	0,4	7	230	81	39
5	2	650	0,2	1,21	0,4	0,4	18	119	52	66
6	1	655	0,2	1,21	0,2	0,4	7	215	75	42
7	1	655	0,2	1,21	0,2	0,4	7	210	70	45
8	1	655	0,2	1,21	0,2	0,4	7	208	68	51
9	2	650	0,2	1,21	0,4	0,4	18	127	61	67
10	1	672	0,22	1,2	0,3	0,35	16	150	59	57

Задание 2. Построить нечеткую модель объекта управления с семью входными x_1 - x_7 и тремя выходными y_1 - y_3 переменными на основе использования математического аппарата нечетких систем в программе *MATLAB 7.0.1*. Оценить точность разработанной модели по выходной переменной y_2 . Исходные данные для построения модели и проверки точности приведены соответственно в таблицах 3 и 4.

Задание 10. Построить нечеткую модель объекта управления с семью входными x_1 - x_7 и тремя выходными y_1 - y_3 переменными на основе использования математического аппарата нечетких систем в программе *MATLAB* 7.0.1. Оценить точность разработанной модели по выходной переменной y_2 . Исходные данные для построения модели и проверки точности приведены соответственно в таблицах 1 и 2.

Лабораторная работа №3

Построение моделей на нейронных сетях

1. Цель работы

Ознакомление магистрантов с методами разработки математических моделей на основе использования нейронных сетей в программе *STATISTICA Neural Networks*.

2. Теоретические сведения

В большинстве случаев нейронные сети используются для предсказания (прогнозирования) значений одной или нескольких зависимых переменных от множества независимых переменных. Задачей временного ряда является предсказание зависимой (более поздней) переменной от ранних значений самой переменной.

В большинстве случаев нейронная сеть используется для предсказания одной переменной. При необходимости предсказания нескольких переменных используют несколько нейронных сетей с одним выходом. Опыт показывает, что сети с одной выходной переменной более предпочтительнее, чем сеть с многими выходными переменными, потому что взаимосвязи между выходными переменными создают затруднения при обучении такой сети.

Для моделирования объекта управления в лабораторной работе в качестве архитектуры нейронной сети выбирается многослойный персептрон, т.к. он позволяет решать широкий круг задач: прогнозирование, классификацию, кластеризацию, распознавание, аппроксимацию. Данная архитектура достаточно проста по структуре и в то же время широко используется для решения прикладных задач [1].

В качестве алгоритма обучения в работе выбран алгоритм обратного распространения. Алгоритм последовательно обучает сеть на данных из обучающего множества, уменьшает суммарную ошибку на всем обучающем множестве [2].

Процесс обучения нейронной сети может рассматриваться как настройка архитектуры и весов связей для эффективного выполнения поставленной задачи. Существуют три парадигмы обучения: *с учителем, без учителя* (самообучение) и *смешанная*.

Метод обучения «с учителем» заключается в последовательном предъявлении примеров значений входных переменных. На каждый входной пример существует требуемый ответ. Веса настраиваются таким образом, чтобы выходы сети были как можно более близкими к требуемым ответам. Более «жесткий» вариант обучения с учителем предполагает, что известна только критическая оценка правильности выхода нейронной сети, а не сами требуемые значения выхода. После многократного предъявления примеров веса сети стабилизируются, причем сеть дает правильные ответы на все (или почти все) примеры из базы данных. В таком случае говорят, что сеть обучена. В программных реализациях можно видеть, что в процессе обучения величина ошибки (сумма квадратов ошибок по всем выходам)

постепенно уменьшается. Когда величина ошибки достигает нуля или приемлемо малого уровня, обучение останавливают, и сеть готова к распознаванию.

Для обучения без учителя не нужно знания требуемых ответов на каждый пример обучающей выборки. В этом случае происходит распределение образцов по категориям (кластерам) в соответствии с внутренней структурой данных или степенью корреляции между образцами.

Нейронная сеть считается устойчивой, если после конечного числа итераций обучения ни один из примеров обучающей выборки не изменяет своей принадлежности в кластерах.

При смешанном обучении весовые коэффициенты одной группы нейронов настраиваются посредством обучения с учителем, а другой группы - на основе самообучения.

Важно отметить, что вся информация, которую сеть приобретает о задаче, содержится в наборе примеров. Поэтому качество обучения сети зависит от количества примеров в обучающей выборке, а также от того, насколько полно эти примеры описывают задачу. Считается, что для полноценной тренировки требуется хотя бы несколько десятков (а лучше сотен) примеров.

Для проверки правильности обучения построенной нейронной сети в нейромиматорах предусмотрены специальные средства ее тестирования. В сеть вводится некоторый сигнал, который, как правило, не совпадает ни с одним из входных сигналов примеров обучающей выборки. Далее анализируется получившийся выходной сигнал сети [6].

Тестирование обученной сети может проводиться либо на одиночных входных сигналах, либо на тестовой выборке, которая имеет структуру, аналогичную обучающей выборке, и также состоит из пар (<вход>, <требуемый выход>). Обычно, обучающая и тестовая выборки не пересекаются. Тестовая выборка строится индивидуально для каждой решаемой задачи.

Одним из недостатков нейронных сетей является явление «переобучения». Оно заключается в том, что ошибки обучения на элементах обучающего множества оказываются очень малыми, но когда на вход сети подаются новые данные, то погрешность существенно возрастает. Это означает, что сеть «заучила» примеры, но не научилась решать задачу. Для преодоления эффекта переобучения используется разбиение исходных данных на три выборки. Обучающее подмножество используется для настройки параметров сети; контрольное – для того, чтобы контролировать, когда в сети появится эффект переобучения; ошибка тестового множества применяется для сравнения различных моделей [3]. Задача определения длины выборок не формализована, решается постановкой вычислительного эксперимента в каждом конкретном случае.

Модели, реализуемые программами-имитаторами нейронов, чрезвычайно разнообразны. Для построения нейронной сети, ориентированной на решение конкретной задачи, используются процедуры формирования нейронных сетей, которые обеспечивают ввод указанных характеристик моделей нейронов и структур нейронных сетей [5]. Каждая группа моделей нейронных сетей может быть использована для решения лишь некоторого ограниченного класса практических задач. Так, многослойные и полносвязные нейронные сети с сигмоидальными передаточными функциями используются для распознавания образов и адаптивного управления; нейронные сети с локальными связями - для обработки изображений и некоторых других частных задач. Для решения задач линейной алгебры используются многослойные сети с особыми передаточными функциями [6].

Лишь для небольшого числа моделей нейронных сетей существует строгое математическое обоснование возможности их применения для решения конкретных практических задач. В наибольшей степени теоретически проработаны двухслойные нейронные сети с сигмоидальными передаточными функциями [6].

Наиболее известный пакет для создания и изучения нейросетевых моделей – *STATISTICA Neural Networks*. Данный программный продукт содержит самые современные и мощные алгоритмы обучения сетей, открытый *API* – интерфейс и встроенные функции Мастер решения задач, а также Генетический алгоритм отбора входных данных.

Пакет *STATISTICA Neural Networks* прост в использовании. Нейронные сети **учатся на примерах**. Пользователь нейронной сети подбирает представительные данные, а затем запускает **алгоритм обучения**, который автоматически воспринимает структуру данных. При этом от пользователя, конечно, требуется какой-то набор эвристических знаний о том, как следует отбирать и подготавливать данные, выбирать нужную архитектуру сети и интерпретировать результаты.

Автоматический конструктор сети самостоятельно выбирает подходящий тип сети и вариант архитектуры (т.е. число скрытых элементов). В стандартном варианте большинство пользователей просто проводят серию экспериментов с сетями различных типов и архитектур, выделяя на эвристическом уровне наиболее многообещающие, после чего именно на них и сосредотачивается все внимание. Автоматический конструктор сети действует по такой же схеме, и даже более формально. Задачу определения архитектуры сети он рассматривает как задачу оптимизации и для исследования возможных вариантов архитектур использует довольно сложные методы поиска (в том числе линейный поиск и вариант метода «отжига»). При этом работа алгоритма может занимать весьма длительное время, поскольку он перебирает множество различных вариантов архитектур. Однако времени у него уйдет не больше, чем у человека, выполняющего ту же серию экспериментов, а преимущество этого метода в том, что в процессе работы он не

требует присутствия пользователя. Кроме того, этот алгоритм осуществляет перебор гораздо большего числа вариантов, чем способен выполнить человек, и как следствие, его результаты часто оказываются лучше.

Одна из наиболее трудных задач при построении нейронной сети – выбор архитектуры, а после того, как архитектура выбрана, - задание ее свободных параметров в большой степени зависит от сложности задачи, которую нужно решить, и поскольку заранее неизвестно, насколько задача сложна, необходим ряд экспериментов. Самое трудное – выбор числа промежуточных слоев и числа элементов в них.

Существуют следующие правила определения промежуточных слоев в многослойном персептроне *MPL*:

- При использовании *MPL* для большинства практических задач достаточно одного промежуточного слоя. Если даже при большом числе скрытых элементов не удастся уменьшить ошибку до приемлемого уровня, можно попробовать сеть с двумя промежуточными слоями.

- Чем больше число скрытых элементов в сети, тем более сложную задачу она может моделировать, но при этом потребуется более длительное обучение и возникнет опасность переобучения. При экспериментировании нужно следить за тем, *как меняются обучающая и контрольная ошибки*. Если при добавлении новых элементов уменьшаются обе ошибки, то, по-видимому, сеть пока слишком мала. Если же контрольная ошибка стала намного больше, чем ошибка обучения (и, в частности, если она увеличивается при дальнейшем обучении), то, вероятно, что сеть слишком большая.

- Общее число весов и пороговых коэффициентов в сети должно быть меньше, а лучше – намного меньше, чем число обучающих наблюдений. В идеале нужно *иметь наблюдений в десять или двадцать раз больше, чем имеющееся количество весов*. Если обучающих наблюдений недостаточно, следует ограничиться сетями небольших размеров, поскольку для моделирования сложной функции попросту недостает данных. *Если число наблюдений меньше, чем число входов, умноженное на число выходов, следует использовать только линейные модели*.

- Для оценки числа нейронов в скрытых слоях однородных нейронных сетей можно воспользоваться формулой (данная формула является следствием из теоремы Колмогорова-Арнольда-Хехт-Нильсена о представимости любой многомерной функции нескольких переменных с помощью нейронной сети фиксированной размерности) [7]. Для оценки необходимого числа синаптических весов W в многослойной сети с сигмоидальными передаточными функциями можно воспользоваться формулой:

$$\frac{mN}{1 + \log_2 N} \leq W \leq m \left(\frac{N}{n} + 1 \right) (n + m + 1) + m \quad (1)$$

где n – размерность входного сигнала, m – размерность выходного сигнала, N – число элементов обучающей выборки.

Оценив необходимое число весов, можно рассчитать число нейронов в скрытых слоях. Например, для двуслойной сети это число составит:

$$L = \frac{W}{n + m} \quad (2)$$

Известны и другие формулы для оценки, например:

$$2(n + L + m) \leq N \leq 10(n + L + m), \quad (3)$$

$$\frac{N}{10} - n - m \leq L \leq \frac{N}{2} - n - m. \quad (4)$$

Точно так же можно рассчитать число нейронов в сетях с большим числом нейронов.

Пример. Имеются статистические данные функционирования объекта управления – технологического процесса производства листового стекла флоат-способом [4]. Данные приведены в файле *steklo.excel*. Используя выборку с 1 по 100 строку построить нейросетевую модель, описывающую зависимость оптических искажений, видимых в проходящем свете, измеренные по методу "Зебра", от значений режимных переменных технологической линии.

Для обучения нейронных сетей используются выборки из 50 обучающих, 25 контрольных и 25 тестовых наблюдений. Контрольное множество данных будет участвовать в кросс-проверке сетей при их обучении. При отсутствии данной проверки, нейронные сети будут слишком близко следовать обучающим данным («чрезмерная подгонка») и воспринимать не столько структуру данных, сколько содержащиеся в них помехи. Кроме того, нейронные сети, которые обучались без кросс-проверки обладают малой способностью к обобщению данных (т.е. показывать хорошие результаты на совершенно новых данных). Тестовое множество данных, которое вообще никак не будет участвовать в обучении сетей, будет использоваться только после того, как будут сформированы окончательные варианты обученных сетей.

Для решения поставленной задачи выберем типовую нейронную сеть - **многослойный персептрон MPL** с прямой передачей сигнала и с сигмоидальными передаточными функциями. Входной слой служит для ввода значений входных переменных. Каждый из скрытых и выходных нейронов соединен со всеми элементами предыдущего слоя. Можно было бы рассмот-

реть сети, в которых нейроны связаны только с некоторыми из нейронов предыдущего слоя; однако, для большинства приложений предпочтительны сети с полной системой связей.

А) Рассмотрим построение нейронной сети, создаваемой автоматическим конструктором, когда число входных переменных определено структурой линейной регрессионной моделью [4]:

$$Zlb=f(Tcm_2, Tza, Tzc, Tolsc)$$

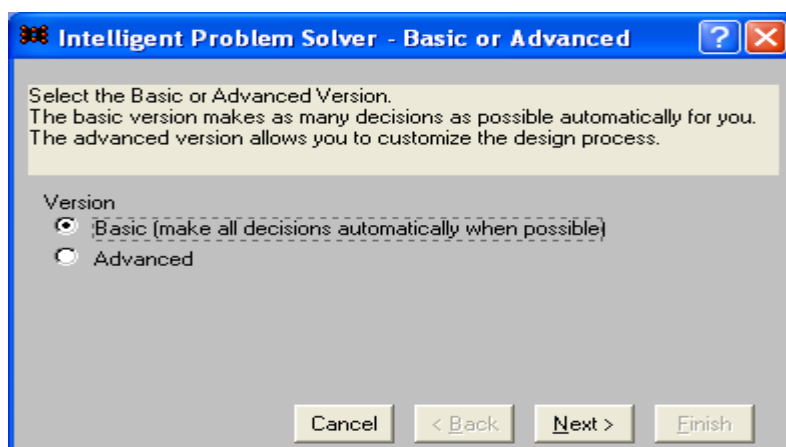
1) Для моделирования нейронной сети вводим данные в пакет *STATISTICA Neural Networks*. Исходные данные для нейронной сети, моделирующей качество вырабатываемого стекла, представлены в следующем виде:

Таблица 1- Статистические данные функционирования объекта управления

VAR1	VAR2	VAR3	VAR4	VAR5	VAR6	VAR 7	VAR 8	VAR9	VAR 10	VAR 11	VAR1 2	VAR 13	VAR 14
<i>Tcm2</i> 1	<i>Tcanal</i> 1	<i>Tin1R</i> 1	<i>Tin12R</i> 1	<i>Tin20R</i> 1	<i>ToutC</i> 1	<i>Tza1</i>	<i>Tzc1</i>	<i>Oxy- gen1</i>	<i>Tolsc</i> 1	<i>Razn1</i> 1	<i>Zlb1</i>	<i>R1dol</i> 1	<i>R2dol</i> 1
<i>Tcm2</i> 2	<i>Tca- nal2</i>	<i>Tin1R</i> 2	<i>Tin12R</i> 2	<i>Tin20R</i> 2	<i>ToutC</i> 2	<i>Tza2</i>	<i>Tzc2</i>	<i>Oxygen</i> 2	<i>Tolsc</i> 2	<i>Razn1</i> 2	<i>Zlb2</i>	<i>R1dol</i> 2	<i>R2dol</i> 2
...								

Таблица содержит выборку из 100 строк с данными функционирования объекта управления. Первые 10 колонок заполнены входными переменными, последние с 11 по 14 колонку содержат выходные переменные. Оптических свойств вырабатываемого стекла, измеренные по методу «Зебра» находятся в колонке VAR12.

2) Создание нейронных сетей автоматическим конструктором начинается с открытия диалогового окна *Basic or Advanced* (рисунок ниже):



Используем второй вариант (*Advanced*) построения нейронных сетей, поскольку он отличается от первого расширенными настройками. Далее нажав клавишу *Next* выбираем тип проблемы *Standart*. Затем выбираем мышкой показатель качества стекла *Zlb*, который

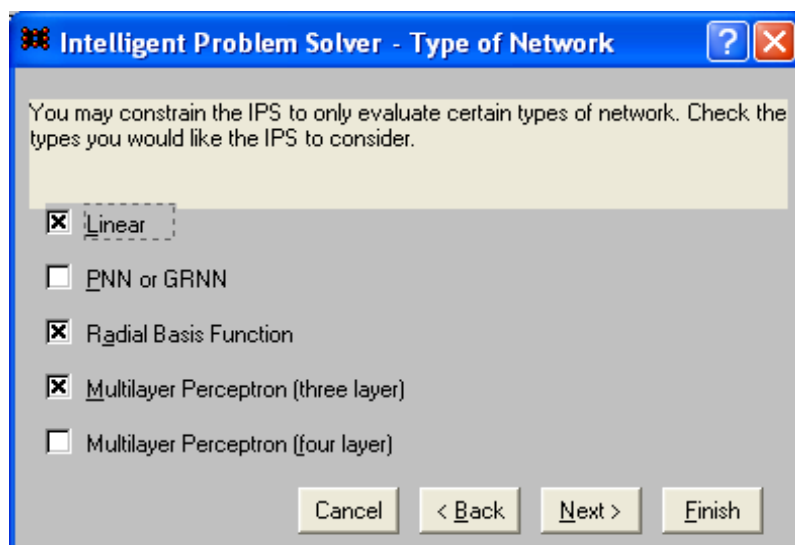
находится в колонке VAR12. Повторно нажимая клавишу *Next* выбираем входные переменные, расположенные в колонках VAR1, VAR7, VAR8, VAR10.

На следующем шаге автоматического конструктора нейронной сети указываем количество обучающих (*Training*) 50, контрольных (*Verification*) 25 и тестовых (*Test*) 25 наблюдений.

Далее в окне *Type of Network* нужно выбрать типы сетей, которые будут анализироваться. Они упорядочены в порядке сложности обучения (рисунок ниже). Линейные (*Linear*) сети почти не требуют обучения и включены сюда потому, что дают хорошую точку отсчета для сравнения эффективности различных методов. Сети *PNN* и *GRNN* также довольно просты, радиальные базисные функции (*Radial Basis Functions*) устроены несколько сложнее, а трех- и четырехслойные многослойные перцептроны (*Multiplayer Perceptrons*) – это сложные конструкции.

Для трехслойного *MPL* поиск сводится к выбору числа элементов в скрытом слое. По существу, это линейный поиск для функции, содержащей помехи (при разных прогонах обучения сети, даже с одним и тем же числом элементов, могут получаться немного различные результаты).

3) На следующем шаге определяем сложность нейронной сети. Сложность нейронной сети зависит от количества нейронов (L) в скрытых слоях, числа синаптических связей (W) в сети и числа элементов обучающей выборки (N). Из расчетов числа нейронов в промежуточном слое имеем следующее:



при выбранном $N=50$, получаем:

$W \leq 82$ по формуле (1),

$L \leq 16$ по формуле (2).

Таким образом, для быстроты анализа автоматическому конструктору в окне *Hidden Units* задаем число синаптических связей 40 и среднее значение числа нейронов в промежуточном слое, равное 8.

4) На следующих этапах задаем время в течение, которого будет идти поиск *Duracionof design process-Timed=10* минут. В окне *Saving Networks* выбираем максимальное число лучших сетей 10, задаем режимы: *Balance performance...* и *Increase...* В следующем окне *Results Shown* помечаем все три окна, обеспечивающие сохранение листинга с результатами, суммарной статистики и с анализом чувствительности лучшей сети.

Затем нажимаем клавишу *Finish*.

5) Автоматический анализатор выдает информацию, содержащуюся в окнах.

- В окне *Intelligent Problem Solver Messages* сообщается о том, что было протестировано 9807 нейронных сетей с различными архитектурами. Определены 10 лучших. Самая лучшая из протестированных сетей обладает следующими характеристиками (коэффициенты регрессии 0.275, корреляции 0.962).

- В окне *Network Set Editor* приводятся структур 10-и лучших сетей с характеристиками: среднеквадратической ошибки, число входных переменных, количество нейронов в скрытом слое, *performance*. В рассматриваемом примере наилучшей выбрана модель типа *MLP* с двумя входными переменными (из заданных четырех переменных), два нейрона в скрытом слое, среднеквадратическая ошибка 3,06, *performance* 0,38.

- В окне *Sensitivity Analysis* приводятся результаты анализа чувствительности нейронной сети к входным данным для обучающей и контрольной выборки. Переменные ранжированы по чувствительности. В рассматриваемом примере видно, что нейронная сеть, описывающая оптические искажения, видимые в проходящем свете, измеренные по методу "Зебра, наиболее чувствительна к изменению значений режимных переменных *VAR10* (*Tolsc* толщина ленты стекла) и *VAR1* (*Tcm₂* температура стекломассы в стекловаренной печи).

	VAR1	VAR10
Rank	2	1
Error	4.201558	7.250218
Ratio	1.825048	3.149307
Rank	2	1
Error	4.347205	7.871148
Ratio	1.420043	2.571163

Обозначения:

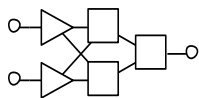
Error – значение среднеквадратической ошибки сети в случае если рассматриваемая переменная не была бы в входных данных;

Ratio – отношение значения *Error* к значению *Baseline Error*

Baseline Error - значение среднеквадратичной ошибки сети в случае, если все переменные (их было четыре), будут входными данными 2, 3 3, 06

Черным цветом помечены результаты на обучающей выборке, красным - на контрольной.

- Ниже показана структура лучшей нейронной сети типа *MLP*, прогнозирующей оптические свойств вырабатываемого стекла, измеряемых по методу «Зебра».



Оценка качества данной нейронной сети представлена в диалоговом окне (Регрессионная статистика).

- В окне *Regression Statistics* (статистики регрессии, рисунок 1) подсчитываются среднее *Data Mean* и стандартное отклонение *Data S.D.* выходной переменной, среднее значение *Error Mean* и стандартное отклонение *Error S.D.* ошибки сети, а также отношение стандартного отклонения ошибки к стандартному отклонению данных *S.D. Ratio* и коэффициент корреляции между спрогнозированным и фактическими значениями *Correlation* на обучающей, контрольной и тестирующих выборках. Обычно значение средней ошибки (*Error Mean*) получается близким к нулю, а стандартное отклонение прогноза оказывается существенно меньше стандартного отклонения данных.

Если величина *S.D. Ratio* меньше 0,1, это означает хорошее качество регрессии.

	Tr. VAR12	Ve. VAR12	Te. VAR12
Data Mean	50.8	52.2	50.04
Data S.D.	7.920524	8.170067	7.103286
Error Mean	0.3565578	-0.3496	1.419502
Error S.D.	2.297474	3.104001	2.729342
Abs E. Mean	1.881901	2.443777	2.307839
S.D. Ratio	0.290066	0.3799235	0.3842365
Correlation	0.9571925	0.9258903	0.9347443

Рисунок 1 – Окно *Regression Statistics*

RMS Error Train		2.3	Verify	3.06	Test	3.0
	VAR12	TVAR12	EVAR12	Error		
01	66.90376	66	0.9037552	0.03013		
02	64.86102	67	-2.138984	0.0713		
03	64.73468	66	-1.265321	0.04218		
04	64.70245	63	1.702451	0.05675		
05	64.70245	65	-0.297549	0.0099183		
06	64.70245	65	-0.297549	0.0099183		
07	64.73468	62	2.734679	0.09116		
08	64.73468	64	0.7346787	0.02449		
09	64.73468	64	0.7346787	0.02449		
10	60.06796	60	0.06796	0.002265		
11	60.13151	61	-0.868486	0.02895		
12	54.58704	52	2.587039	0.08623		
13	55.00663	55	0.006633	0.0002211		
14	54.92309	56	-1.076908	0.0359		
15	49.5371	51	-1.46290	0.04876		
16	49.44996	50	-0.550036	0.01833		
17	49.71162	49	0.7116171	0.02372		
18	45.26036	48	-2.739643	0.09132		

Рисунок 2 - Окно *Run Data Set*

- В окне *Run Data Set* (прогнать набор данных) содержатся следующие значения (перечисленные слева направо): фактические выходы сети, целевые выходные значения, ошибки, т.е. разности между первым и вторым, и суммарная ошибка по каждому наблюдению. Над таблицей выдается итоговая среднеквадратичная ошибка (СКО) – *RMS error* сети на наборе: обучающих -50, контрольных -25, тестовых -25 данных (Рисунок 2).

Б) Рассмотрим построение нейронной сети, создаваемых автоматическим конструктором, когда входные переменные не конкретизированы и заданы множеством всех 10-и входных переменных.

Создание нейронных сетей автоматическим конструктором начинается с открытия диалогового окна *Basic or Advanced*. Помечаются мышкой выходная переменная VAR12 и все 10 входных переменных VAR1 - VAR10.

Автоматический анализатор выдает информацию, содержащуюся в окнах.

- В окне *Intelligent Problem Solver Messages* сообщается о том, что было протестировано 2839 нейронных сетей с различными архитектурами. Определены 10 лучших. Самая лучшая из протестированных сетей обладает следующими характеристиками (коэффициенты регрессии 0.2, корреляции 0.98).

- В окне *Network Set Editor* приводятся структур 10-и лучших сетей с характеристиками: среднеквадратической ошибки, число входных переменных, количество нейронов в скрытом слое, *performance*. В рассматриваемом примере наилучшей выбрана модель типа *MLP* с 9-ю входными переменными, 7 нейронов в скрытом слое, среднеквадратическая ошибка 1,75, *performance* 0,2.

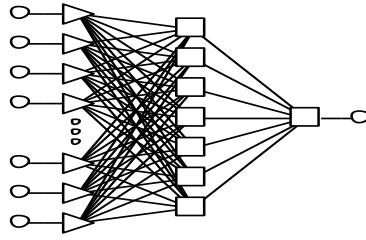
- В окне *Sensitivity Analysis* (рисунок 4.3) приводятся результаты анализа чувствительности нейронной сети к входным данным для обучающей и контрольной выборки.

	VAR1	VAR2	VAR3	VAR4	VAR6	VAR7	VAR8	VAR9	VAR10
Rank	4	3	2	7	9	5	6	8	1
Error	2.43	2.96	3.17	1.96	1.48	2.43	2.25	1.50	6.60
Ratio	1.91	2.33	2.49	1.54	1.16	1.91	1.77	1.18	5.18
Rank	3	5	2	6	9	4	7	8	1
Error	3.06	2.88	3.49	2.85	1.69	3.02	2.68	2.17	9.90
Ratio	1.74	1.64	1.98	1.62	0.96	1.72	1.53	1.24	5.64

Рисунок 3 - Окно *Sensitivity Analysis*

Входные переменные ранжированы по чувствительности. В рассматриваемом примере видно, что нейронная сеть, описывающая оптические свойства вырабатываемого стекла по методу «Зебра», наиболее чувствительна к изменению значений режимных переменных VAR10 (*Tolsc* толщина ленты стекла), VAR3 (*Tin1R* температура в первом пролете флотванны), VAR2 (*Tcanal* температура стекломассы в выработочном канале стекловаренной печи), VAR1 (*Tcm2* температура стекломассы в стекловаренной печи).

- Ниже показана структура лучшей нейронной сети типа *MLP*, прогнозирующей оптические свойств вырабатываемого стекла, измеряемых по методу «Зебра».



- В окне *Regression Statistics* (рисунок 4) приводятся среднее *Data Mean* и стандартное отклонение *Data S.D.* выходной переменной, среднее значение *Error Mean* и стандартное отклонение *Error S.D.* ошибки сети, а также отношение стандартного отклонения ошибки к стандартному отклонению данных *S.D. Ratio*, коэффициент корреляции между спрогнозированным и фактическими значениями *Correlation* на обучающей, контрольной и тестирующих выборках.

	Tr.VAR12	Ve.VAR12	Te.VAR12
Data Mean	49.68	53.72	50.76
Data S.D.	5.902058	8.715503	9.452866
Error Mean	-0.0022	-0.02658	0.4997012
Error S.D.	1.286572	1.790656	3.536783
Abs E. Mean	0.9531708	1.242784	2.74766
S.D. Ratio	0.2179871	0.2054564	0.3741493
Correlation	0.9759938	0.9816402	0.928383

Рисунок 4 – Окно *Regression Statistics*

- В окне *Run Data Set* содержатся фактические значения выходных переменных, выходные переменные сети, ошибки сети и суммарная ошибка по каждому наблюдению. Над

	RMS Error Train	1.2	Verify	1.75	Test	3.5
	VAR12	TVAR12	EVAR12	Error		
01	73.11785	66	7.11785	0.2542089		
02	62.75665	67	-4.243352	0.1515483		
03	62.59275	66	-3.40725	0.1216875		
04	62.76189	63	-0.238112	0.008504		
05	64.70936	65	-0.2906	0.01038		
06	65.56636	65	0.5663574	0.02023		
07	62.88834	62	0.8883444	0.03173		
08	62.68308	64	-1.316916	0.04703		
09	64.40124	64	0.401244	0.01433		
10	56.18116	60	-3.81884	0.1363872		
11	60.72658	61	-0.273418	0.009765		
12	53.01855	52	1.018553	0.03638		
13	54.7895	55	-0.2105	0.007518		
14	57.35329	56	1.353293	0.04833		
15	50.4636	51	-0.5364	0.01916		
16	50.59852	50	0.5985198	0.02138		
17	49.24055	49	0.2405537	0.008591		
18	44.49586	48	-3.504142	0.1251479		

таблицей выдается итоговая среднеквадратичная ошибка *RMS error* сети на наборе: обучающих -50, контрольных -25, тестовых -25 данных (Рисунок 5).

Сравнительный анализ характеристик нейронных сетей, основанных на регрессионных моделях и характеристик нейронных сетей,

Рисунок 5 - Окно *Run Data Set*

созданных с автоматическим определением значимых входных переменных, показал, что последние нейронные сети обладают более лучшими характеристиками прогнозирования т.е. имеют более низкий показатель *Regression Ratio* (отношение стандартного отклонения ошибки к стандартному отклонению данных) и более высокий коэффициент корреляции между прогнозируемыми и фактическими значениями (Таблица 2).

Таблица 2 – Сравнительные характеристики нейронных сетей при моделировании оптических свойств вырабатываемого стекла, измеряемых по методу «Зебра»

Тип нейронной сети	Количество нейронов			Характеристики сети		
	на входе	в промежуточном слое	на выходе	Коэффициент регрессии	Коэффициент корреляции	Среднеквадратическая ошибка
<i>MLP</i>	2	2	1	0,27	0,96	3,06
<i>MLP</i>	9	7	1	0,2	0,98	1,75

В качестве информационной модели процесса формования ленты стекла на расплаве олова выбираем нейронную сеть *MLP* с числом нейронов на входе - 9, в промежуточном слое - 7, на выходе -1. Фактические данные изменения оптических искажений, видимых в проходящем свете, измеренные по методу "Зебра» VAR12 и результаты моделирования *T.VAR12* приведены на рисунке 6.

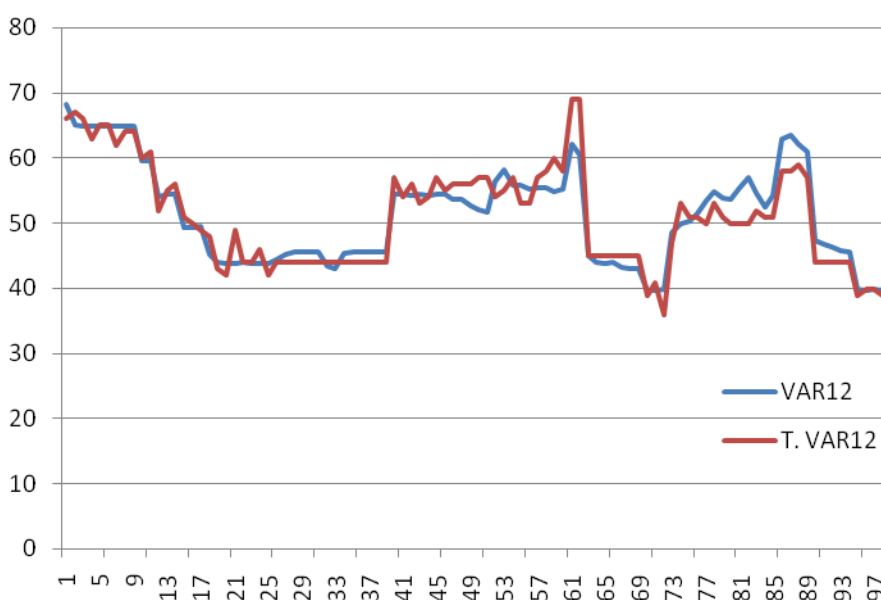


Рисунок 4.6 - Фактические данные изменения оптических искажений, видимых в проходящем свете, измеренные по методу "Зебра" VAR12 и результаты моделирования *T.VAR12*

3. Задание к лабораторной работе

Разработать математическую модель объекта управления на основе использования нейронных сетей в программе *STATISTICA Neural Networks*. Оценить адекватность разработанной модели на обучающей, а точность на контрольной выборке. Составить отчет по лабораторной работе и сделать выводы по результатам исследований.

4. Порядок выполнения лабораторной работы

Получить от преподавателя индивидуальное задание. Ознакомиться с методом построения нейронных сетей в программе *STATISTICA Neural Networks*.

1) Создать в программе *STATISTICA Neural Networks* таблицу с исходными данными путем копирования данных из файла *steklo.exe*.

2) Построить нейронную сеть с помощью автоматического конструктора при заданном числе входных переменных, определяемом структурой линейной регрессионной модели [4].

3) Проанализировать структуру разработанной нейронной сети.

4) Построить второй вариант нейронной сети с помощью автоматического конструктора с использованием 10-и входных переменных.

5) Оценить качество двух разработанных нейронных сетей и их пригодность для математического описания объекта управления по величине среднеквадратичной ошибки модели и коэффициенту парной корреляции между фактическими и модельными данными. Выбрать лучшую нейронную сеть.

6) Построить совмещенные графики изменения выходной переменной с использованием фактических и модельных данных.

7) Сделать выводы по выполненной лабораторной работе.

5. Содержание отчета

1) Индивидуальное задание на лабораторную работу.

2) Нейронная сеть, построенная с помощью автоматического конструктора при заданном числе входных переменных, определяемом структурой линейной регрессионной модели. Структура сети и статистические характеристики.

3) Второй вариант нейронной сети, построенный с помощью автоматического конструктора с использованием 10-и входных переменных. Структура сети и статистические характеристики.

4) Результаты сравнительной оценки качества разработанных нейронных сетей по величине среднеквадратичной ошибки модели и коэффициенту парной корреляции между фактическими и модельными данными.

5) Обоснование выбора лучшей нейронной сети для математического описания объекта управления.

6) Совмещенные графики изменения выходной переменной с использованием фактических и модельных данных.

7) Выводы по результатам лабораторной работы.

6. Вопросы для самоконтроля

1) Назначение нейронных сетей, область их применения.

2) Процесс обучения нейронной сети. Методы обучения сетей, их особенности.

3) От чего зависит качество обучения нейронной сети?

4) «Переобучение» нейронной сети. Преодоление эффекта переобучения.

5) Назначение тестирования нейронной сети. Технология тестирования обученной сети.

6) Модели, реализуемые программами-имитаторами нейронов. Области использования многослойных и полносвязных нейронных сетей с сигмоидальными передаточными функциями, нейронных сетей с локальными связями, многослойных сетей с особыми передаточными функциями.

7) Особенности пакета *STATISTICA Neural Networks*. Назначение пакета и его содержание.

8) Правила определения промежуточных слоев в многослойном персептроне *MPL*: оценка числа нейронов в скрытых слоях однородных нейронных сетей.

9) Метод построения нейронной сети с помощью автоматического конструктора при заданном числе входных переменных.

10) Метод построения нейронной сети с помощью автоматического конструктора, когда входные переменные не конкретизированы и заданы множеством переменных.

11) Оценка адекватности и точности моделей, построенных на нейронных сетях.

7. Список рекомендуемой литературы

1) Галушкин А.И. Теория нейронных сетей. Кн.1: Учеб. пособие для вузов. – М.: ИПРЖР, 2001. -385 с.

2) Короткий С. Нейронные сети: алгоритм обратного распространения.//<http://www.neuropower/ole/rus/book/index.html>

3) Медведев, В.С. Нейронные сети. *MATLAB 6*/ В.С. Медведев, В.Г. Потемкин; под общ. ред. В.Г. Потемкина. – М.: ДИАЛОГ-МИФИ, 2002. – 496 с. – ISBN 5–86404–163–7

4) Р.И. Макаров, В.В. Тарбеев, Е.Р. Хорошева и др. Управление качеством листового стекла (флоат-способ): Учебное пособие. -М.: Издательство Ассоциации строительных вузов, 2004.-152с.

5) Барский А.Б. Нейронные сети: распознавание, управление, принятие решений. – М.: Финансы и статистика, 2004. – 176с.

6) Борисов Ю, Кашкаров В., Сорокин С. Нейросетевые методы обработки информации и средства их программно-аппаратной поддержки. – Открытые системы. – 1997.-№4.

7) Круглов В.В., Борисов В.В. Искусственные нейронные сети. Теория и практика – 2-е изд. – М.: Горячая линия – Телеком, 2002. – 382 с.

8. Варианты заданий к лабораторной работе

Задание 1. Построить математическую модель объекта управления с десятью входными и четырьмя выходными переменными на основе использования пакета для создания и изучения нейросетевых моделей – *STATISTICA Neural Networks*. Оценить точность разработанной модели по выходной переменной *Zlb*, характеризующей оптические искажения, видимые в проходящем свете, измеренные по методу "Зебра". Исходные данные для построения модели приведены в файле *steklo.exe*. Для построения модели использовать выборку из таблицы функционирования объекта управления, строки с 101 по 200. Регрессионная модель, описывающая «Зебру» имеет следующую структуру: $Zlb=f(Tcm2, Tza, Tzc, Tolsc)$.

Задание 2. Построить математическую модель объекта управления с десятью входными и четырьмя выходными переменными на основе использования пакета для создания и изучения нейросетевых моделей – *STATISTICA Neural Networks*. Оценить точность разработанной модели по выходной переменной *Zlb*, характеризующей оптические искажения, видимые в проходящем свете, измеренные по методу "Зебра". Исходные данные для построения модели приведены в файле *steklo.exe*. Для построения модели использовать выборку из таблицы функционирования объекта управления, строки с 201 по 300. Регрессионная модель, описывающая «Зебру» имеет следующую структуру: $Zlb=f(Tcm2, Tza, Tzc, Tolsc)$.

Задание 3. Построить математическую модель объекта управления с десятью входными и четырьмя выходными переменными на основе использования пакета для создания и изучения нейросетевых моделей – *STATISTICA Neural Networks*. Оценить точность разработанной модели по выходной переменной *Zlb*, характеризующей оптические искажения, видимые в проходящем свете, измеренные по методу "Зебра". Исходные данные для построения модели приведены в файле *steklo.exe*. Для построения модели использовать выборку из таблицы функционирования объекта управления, строки с 264 по 365. Регрессионная модель, описывающая «Зебру» имеет следующую структуру: $Zlb=f(Tcm_2, Tza, Tzc, Tolsc)$.

Задание 4. Построить математическую модель объекта управления с десятью входными и четырьмя выходными переменными на основе использования пакета для создания и изучения нейросетевых моделей – *STATISTICA Neural Networks*. Оценить точность разрабо-

танной модели по выходной переменной $R1dol$, характеризующей оптические искажения, видимые в отраженном свете, измеренные в первой долянке. Исходные данные для построения модели приведены в файле *steklo.exe*. Для построения модели использовать выборку из таблицы функционирования объекта управления, строки с 1 по 100. Регрессионная модель, описывающая $R1dol$ имеет следующую структуру: $R1dol = f(Tcanal, Tin1R, Tin20R, ToutC)$.

Задание 5. Построить математическую модель объекта управления с десятью входными и четырьмя выходными переменными на основе использования пакета для создания и изучения нейросетевых моделей – *STATISTICA Neural Networks*. Оценить точность разработанной модели по выходной переменной $R1dol$, характеризующей оптические искажения, видимые в отраженном свете, измеренные в первой долянке. Исходные данные для построения модели приведены в файле *steklo.exe*. Для построения модели использовать выборку из таблицы функционирования объекта управления, строки с 101 по 200. Регрессионная модель, описывающая $R1dol$ имеет следующую структуру: $R1dol = f(Tcanal, Tin1R, Tin20R, ToutC)$.

Задание 6. Построить математическую модель объекта управления с десятью входными и четырьмя выходными переменными на основе использования пакета для создания и изучения нейросетевых моделей – *STATISTICA Neural Networks*. Оценить точность разработанной модели по выходной переменной $R1dol$, характеризующей оптические искажения, видимые в отраженном свете, измеренные в первой долянке. Исходные данные для построения модели приведены в файле *steklo.exe*. Для построения модели использовать выборку из таблицы функционирования объекта управления, строки с 201 по 300. Регрессионная модель, описывающая $R1dol$ имеет следующую структуру: $R1dol = f(Tcanal, Tin1R, Tin20R, ToutC)$.

Задание 7. Построить математическую модель объекта управления с десятью входными и четырьмя выходными переменными на основе использования пакета для создания и изучения нейросетевых моделей – *STATISTICA Neural Networks*. Оценить точность разработанной модели по выходной переменной $R2dol$, характеризующей оптические искажения, видимые в отраженном свете, измеренные во второй долянке. Исходные данные для построения модели приведены в файле *steklo.exe*. Для построения модели использовать выборку из таблицы функционирования объекта управления, строки с 1 по 100. Регрессионная модель, описывающая $R2dol$ имеет следующую структуру: $R2dol = f(Tcanal, Tin1R, Tin20R, ToutC)$.

Задание 8. Построить математическую модель объекта управления с десятью входными и четырьмя выходными переменными на основе использования пакета для создания и изучения нейросетевых моделей – *STATISTICA Neural Networks*. Оценить точность разработанной модели по выходной переменной $R2dol$, характеризующей оптические искажения, видимые в отраженном свете, измеренные во второй долянке. Исходные данные для построения модели приведены в файле *steklo.exe*. Для построения модели использовать выборку из

таблицы функционирования объекта управления, строки с 101 по 200. Регрессионная модель, описывающая $R2dol$ имеет следующую структуру: $R2dol = f(Tcanal, Tin1R, Tin20R, ToutC)$.

Задание 9. Построить математическую модель объекта управления с десятью входными и четырьмя выходными переменными на основе использования пакета для создания и изучения нейросетевых моделей – *STATISTICA Neural Networks*. Оценить точность разработанной модели по выходной переменной $R2dol$, характеризующей оптические искажения, видимые в отраженном свете, измеренные во второй долянке. Исходные данные для построения модели приведены в файле *steklo.exe*. Для построения модели использовать выборку из таблицы функционирования объекта управления, строки с 201 по 300. Регрессионная модель, описывающая $R2dol$ имеет следующую структуру: $R2dol = f(Tcanal, Tin1R, Tin20R, ToutC)$.

Задание 10. Построить математическую модель объекта управления с десятью входными и четырьмя выходными переменными на основе использования пакета для создания и изучения нейросетевых моделей – *STATISTICA Neural Networks*. Оценить точность разработанной модели по выходной переменной $Razn1$, характеризующей разнотолщинность ленты стекла в первой долянке. Исходные данные для построения модели приведены в файле *steklo.exe*. Для построения модели использовать выборку из таблицы функционирования объекта управления, строки с 1 по 100. Регрессионная модель, описывающая $Razn1$ имеет следующую структуру:

$$Razn1 = f(Tin1R, Tin12R, Tin20R, Tolsc).$$

Задание 11. Построить математическую модель объекта управления с десятью входными и четырьмя выходными переменными на основе использования пакета для создания и изучения нейросетевых моделей – *STATISTICA Neural Networks*. Оценить точность разработанной модели по выходной переменной $Razn1$, характеризующей разнотолщинность ленты стекла в первой долянке. Исходные данные для построения модели приведены в файле *steklo.exe*. Для построения модели использовать выборку из таблицы функционирования объекта управления, строки с 101 по 200. Регрессионная модель, описывающая $Razn1$ имеет следующую структуру:

$$Razn1 = f(Tin1R, Tin12R, Tin20R, Tolsc).$$

Задание 12. Построить математическую модель объекта управления с десятью входными и четырьмя выходными переменными на основе использования пакета для создания и изучения нейросетевых моделей – *STATISTICA Neural Networks*. Оценить точность разработанной модели по выходной переменной $Razn1$, характеризующей разнотолщинность ленты стекла в первой долянке. Исходные данные для построения модели приведены в файле *steklo.exe*. Для построения модели использовать выборку из таблицы функционирования

объекта управления, строки с 201 по 300. Регрессионная модель, описывающая $Razn1$ имеет следующую структуру:

$$Razn1 = f(Tin1R, Tin12R, Tin20R, Tolsc).$$

Лабораторная работа №4
Ситуационное моделирование в системах управления
с активным элементом (человеком).

1. Цель работы

Изучение метода ситуационного управления, возникшего в связи с необходимостью моделирования процессов принятия решений в системах с активным элементом (человеком).

2. Общие сведения

Задача принятия решений трактуется как задача поиска такого разбиения множества ситуаций на классы, при котором каждому классу соответствовало решение, наиболее целесообразное с позиции заданных критериев функционирования. При наличии такого разбиения поиск решения в конкретной ситуации сводился к поиску класса и соотнесения ему решения по управлению. Такая постановка задачи справедлива для систем управления, в которых число потенциально возможных ситуаций (ПВС) существенно превышает (иногда на несколько порядков) число возможных решений по управлению. Этот случай соответствует контекстно-независимому способу вывода решений, когда все множество ПВС разбивается на классы таким образом, чтобы каждому классу в соответствие ставилось решение по управлению [1].

Рассмотрим методику разработки ситуационной модели на основе анализа и формализации действий ЛПР на примере управления технологическим процессом варки-выработки листового стекла [2]. Отклонения режимных переменных рассматриваются как появление ситуаций (начальных) s_n на объекте управления, а нахождение режимных переменных в пределах допусков как конечную ситуацию s_k . Управление можно представить как перевод объекта из начальной ситуации $s_n \in S_n$ в конечную ситуацию $s_k \in S_k$ под действием управляющих воздействий:

$$u_1 \\ s_n \rightarrow s_k .$$

Наличие возможности классификации ситуаций в соответствии с типовыми решениями по управлению, позволяет использовать машинную процедуру формирования понятий *CLS-9* для построения ситуационной модели управления [3]. Понятия представляются в виде деревьев классификаций, устанавливающих соответствие между дискретными представлениями произвольных ситуаций и выбираемыми управляющими воздействиями. Действия ЛПР формализуются на основе высказываний экспертов о принимаемых управляющих решениях в тех или иных технологических ситуациях, возникающих на объекте управления.

Формирование понятий может рассматриваться как задача распознавания образов, когда по признакам (ситуации) определяется значение управляющего воздействия, т.е. из множества критериальных классов управляющего воздействия выбирается нужный класс.

Совокупность всевозможных признаков зависит от количества переменных, характеризующих признаки и числа возможных значений каждого признака. При трех значениях каждого признака $V_i = 3$, учитывающих знак отклонения переменной от задаваемого интервала:

“-1” - выход за пределы нормы в сторону уменьшения (меньше);

“0” - значение переменной в допустимых пределах (норма);

“+1” - выход за пределы нормы в сторону увеличения (больше)

и числе переменных ситуаций n , генеральная совокупность признаков будет иметь размерность 3^n . При этом каждая ситуация будет кодироваться n разрядным (троичным) числом.

Множество критериальных классов для каждого управляющего воздействия ограничивается тремя элементами:

$$K = \{ -, =, + \},$$

где “-” - отрицательное приращение управляющего воздействия (уменьшить);

“ = ” - нулевое приращение (не изменять);

“ + ” - положительное приращение (увеличить).

Процедура *CLS-9* формирует деревья по каждому управляющему воздействию. От вершин деревьев, характеризующих признак, отходят V_i ветвей. Алгоритм предусматривает использование в качестве критерия того признака, который будет наиболее полезным при классификации относящихся к данной вершине объектов. Для описания алгоритма формирования понятий вводится следующая система обозначений:

$A = \{A_j\}$ - множество признаков, не рассматривавшихся ни в одной из вершин, расположенных выше исследуемой;

$K = \{k\}$ - множество критериальных классов;

V_i - количество возможных значений признака i ;

$n_{i,j,k}$ - количество относящихся к рассматриваемой вершине объектов, которые характеризуются значением j признака i и принадлежат к классу k ;

n_{ij}^* - максимальное значение параметра $\{n_{ijk}\}$ для любого из k классов.

Процедура *CLS-9* определяет все члены множества $\{n_{ijk}\}$ и для каждого признака находит значение величины

$$H_i = \sum_{j=1}^{V_i} n_{ij}^* .$$

В качестве критерия выбирается тот признак, для которого значение H_i оказывается максимальным. Затем алгоритм формирует вершину, от которой вниз отходят ветви по числу V_i возможных значений выбранного признака i . Построенное дерево является “наилучшим из возможных” в смысле классификации всех объектов, описания которых были введены в память ЭВМ к моменту формирования дерева. Классификация всех объектов оказывается правильной.

Пример. Дана выборка для формирования понятий по выбору управляющих воздействий для процесса стекловарения в ванной печи средней производительности (таблица 1) [2].

Таблица 1- Выборка для разработки ситуационной модели управления процессом стекловарения

Ситуация	Признаки							Принимаемые решения по управлению		
	$\Theta_{гс}$	$\Theta_{см}$	$Q_{г}$	P	$L_{ш}$	$L_{п}$	M	μ_2	$\mu_в$	$\mu_{ш}$
1	+1	-1	0	0	0	0	0	=	-	=
2	+1	-1	0	+1	-1	0	0	=	+	-
3	+1	-1	0	-1	-1	-1	0	=	=	+
4	+1	-1	0	1	-1	+1	0	=	-	-
5	+1	-1	-1	+1	0	0	-1	+	=	-
5	+1	-1	+1	+1	-1	0	0	=	=	-
7	-1	-1	+1	+1	-1	0	0	=	+	-
8	-1	-1	-1	+1	-1	0	-1	+	+	-
9	+1	-1	-1	-1	-1	0	-1	+	=	+
10	+1	-1	+1	+1	-1	0	-1	=	=	-
11	+1	-1	-1	0	-1	-1	-1	+	=	=
12	+1	-1	+1	+1	-1	-1	0	=	=	-
13	+1	-1	-1	-1	-1	-1	-1	+	=	+
14	-1	-1	-1	-1	-1	-1	-1	+	+	+
15	+1	-1	+1	+1	-1	+1	-1	=	-	-
16	+1	-1	-1	-1	-1	+1	-1	+	-	=

Обозначения признаков: $\Theta_{гс}$ – температура газовой среды в ванной печи; $\Theta_{см}$ – температура стекломассы в ванной печи; $Q_{г}$ – расход газа на ванную стекловаренную печь; P – давление газового пространства в ванной печи; $L_{ш}$ – граница варочной шихты в ванной печи; $L_{п}$ – граница варочной пены в ванной печи; M – число работающих машин вертикального вытягивания стекла.

Обозначения управляющих воздействий: μ_2 – расход природного газа на горелки; $\mu_в$ – расход воздуха на горение; $\mu_{ш}$ – положение шиберы дымовой трубы (тяга дымовых газов).

Пример. Рассмотрим построение ситуационной модели процесса стекловарения с использованием программы *Operator24Final.exe* [4].

Программа предназначена для работы в операционных системах семейства *Windows*.

Программа требует:

- 1 Мб места на жестком диске;
- до 5 Мб свободной оперативной памяти (без учета других запущенных приложений) при работе с не более 50 признаками, не более 50 управлениями и не более 100 ситуациями;
- 500 МГц процессор фирмы *Intel* или *AMD* для работы с небольшими выборками, 2500 МГц для работы с большими выборками.

Процесс функционирования СППР состоит из двух основных алгоритмов:

1. работа в режиме обучения – в этом режиме происходит обучение и настройка программы.
2. работа в режиме диалога – в этом режиме оператор может получать от программы рекомендации по управлению.

Настройка программы

Для настройки в программе предусмотрена специальная вкладка «Настройки обучения» (рисунок 1). Порядок настройки следующий:

1. Внести значения в поля «Количество признаков», «Количество управляющих воздействий», «Количество классов признаков», «Количество классов воздействий», «Объем обучающей выборки». Количества признаков и управляющих воздействий (в версии 2.4 ограничены 20-ю, объем обучающей выборки ограничен 100 значениями).

2. Нажать кнопку «Задать параметры выборки» - новые значения сохранятся в программе. Изменяются поля «Имена признаков и управляющих воздействий» и «Обозначения классов»

3. Если требуется, внести новые имена признаков и управляющих воздействий в соответствующем поле. Имена должны быть не пустыми и уникальными. Эти имена будут использованы для обучения и для выдачи советов. Рекомендуется давать такие имена управляющих воздействий, которые будут легко понятны оператору.

4. Нажать кнопку «Задать» в данном блоке для сохранения введенных имен.

5. Если требуется внести новые обозначения классов признаков и управляющих воздействий. Обозначения должны быть не пустыми и уникальными. Они используются для обучения программы.

Рекомендуется использовать следующие обозначения для классов признаков:

- 1 – для обозначения класса меньше нормы;
- +1 – для обозначения класса больше нормы;
- 0 – для обозначения нормального диапазона.

Рекомендуется использовать следующие обозначения для классов управляющих воздействий:

- – для обозначения класса уменьшения воздействия;
- + – для обозначения класса увеличения воздействия;
- = – для обозначения класса «не изменять».

6. Нажать кнопку «Задать» в данном блоке для сохранения введенных классов.

В версии 2.4 изменение количества и обозначений классов отключено.

После этого процесс настройки закончен, можно переходить к обучению программы.

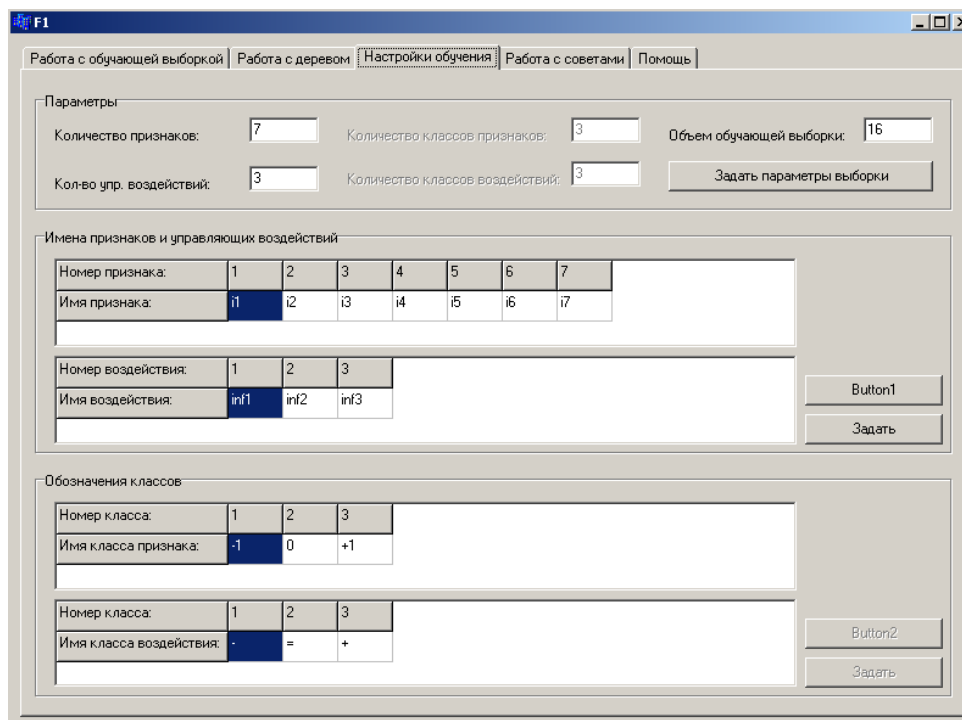


Рисунок 1- Вкладка настроек

Работа в режиме обучения

Режим обучения позволяет создать в памяти программы модель, отражающую поведение объекта. Последовательность обучения следующая:

1. Подготовка обучающей выборки.

- из обучающей выборки должны быть исключены повторяющиеся ситуации;
- значения переменных в обучающей выборке должны быть представлены в виде классов, классы должны соответствовать введенным во время настройки.

2. Занесение обучающей выборки в программу. Для этого нужно перейти на вкладку «Работа с обучающей выборкой» (рисунок 2). В таблице признаков в строки заносятся ситуации, в столбцы – переменные. В таблице управляющих воздействий в строках находятся управления, соответствующие ситуациям (соответствие по номеру).

3. Тестирование правильности внесения выборки производится кнопкой «Тестирование введенной выборки». При этом в соседнем окне будет выведен отчет о проведении тестирования. Если все ячейки заполнены верно, будет выдано сообщение о том, что ошибок нет. В противном случае программа выдаст номера ячеек, в которых обнаружены ошибки.

Начинать строить дерево можно только в том случае, если в таблицах нет ошибок.

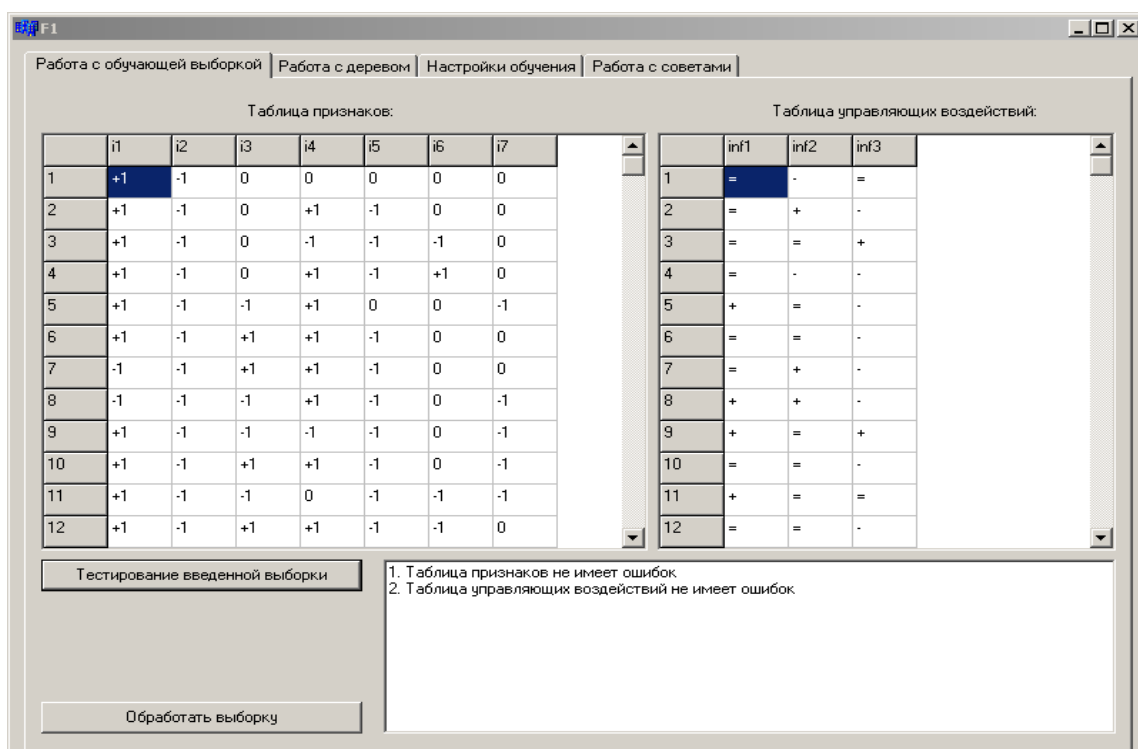


Рисунок 2– Окно работы с обучающей выборкой

4. Далее необходимо перейти на вкладку «работа с деревом» (Рисунок 3).

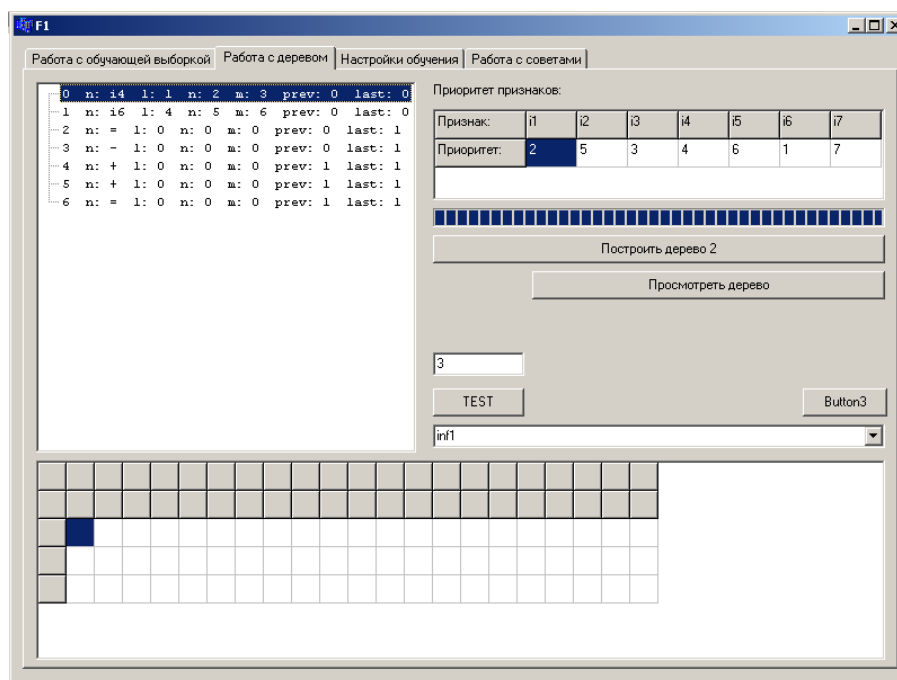


Рисунок 3 – Окно работы с деревом

5. Заполнить таблицу «Приоритет признаков». Она должна быть заполнена числами, которые могут повторяться. Приоритет признаков применяется в том случае, если внутренний алгоритм программы не позволяет выбрать предпочтительный признак. Меньшее число соответствует более высокому приоритету. Более высокий приоритет следует отдавать

наиболее информативным признакам (т.е. если имеются сведения, что некоторый признак несет больше информации об объекте, чем другие). Таблицу можно заполнить одинаковыми значениями, если никакой дополнительной информации о признаках нет. В этом случае будет выбираться первый по порядку.

6. Нажать кнопку «Построить дерево». Эта функция создает модель объекта в памяти. После этого можно переходить к получению советов по управлению.

На вкладке «Работа с деревом» так же имеется возможность просмотреть построенные деревья по каждому управляющему воздействию. Для этого нужно:

- выбрать номер управляющего воздействия в поле рядом с кнопкой «Просмотреть машинное представление дерева»
- нажать кнопку «Просмотреть машинное представление дерева»

Представление дерева появится в окне слева. Расшифровка строк машинного представления дерева представлена на рисунке 4.

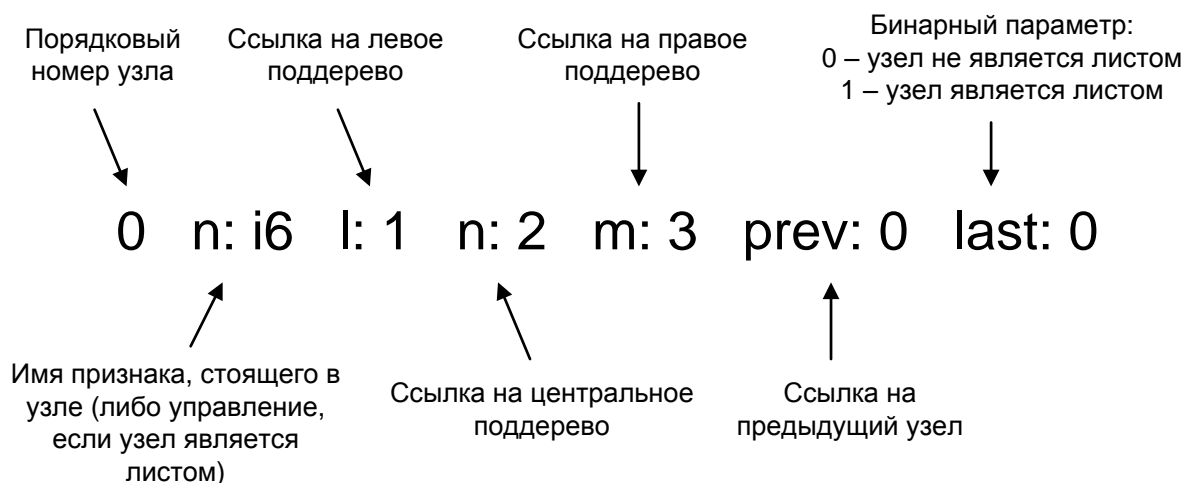


Рисунок 4 – Расшифровка машинного представления дерева

После установки приоритета признакам выполнить команду «Построить дерево». Так же здесь можно посмотреть построенное программой дерево, выполнив команду «Просмотреть дерево». Дерево выводится в машинном представлении, т.е. строками.

По указанной выборке программой было сформировано три дерева. Для примера приведем дерево для управляющего воздействия «поворот шибера дымовой трубы $\mu_{ш}$ », помеченного в программе идентификатором *inf 3*.

0	n: i4	l: 1	n: 2	m: 3	prev: 0	last: 0
1	n: i6	l: 4	n: 5	m: 6	prev: 0	last: 0
2	n: =	l: 0	n: 0	m: 0	prev: 0	last: 1
3	n: -	l: 0	n: 0	m: 0	prev: 0	last: 1
4	n: +	l: 0	n: 0	m: 0	prev: 1	last: 1
5	n: +	l: 0	n: 0	m: 0	prev: 1	last: 1
6	n: =	l: 0	n: 0	m: 0	prev: 1	last: 1

Рисунок 5 – Машинное представление дерева для *inf 3*

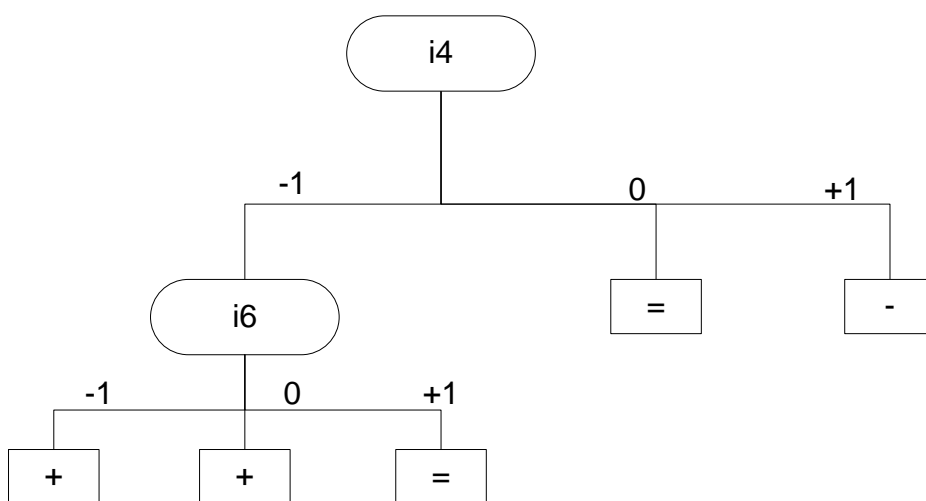


Рисунок 6 – Схема дерева для *inf 3*, построенная вручную по машинному представлению

Работа в режиме диалога (в режиме получения совета)

Для получения советов необходимо выполнить следующие действия:

1. Ввести ситуацию в таблицу.
2. Нажать кнопку «Получить совет»

После этого в поле «Совет» появятся рекомендации по управлению объектом (рисунок 7).

При необходимости введения добавочной ситуации для обучения или замены ранее введенной ситуации новой необходимо вернуться в режимы настройки и обучения. Скорректировать объем обучающей выборки в режиме настройки и ввести дополнительные строки с новыми ситуациями в режим обучения. Необходимо перестроить деревья классификации ситуаций и выполнить тестирование.

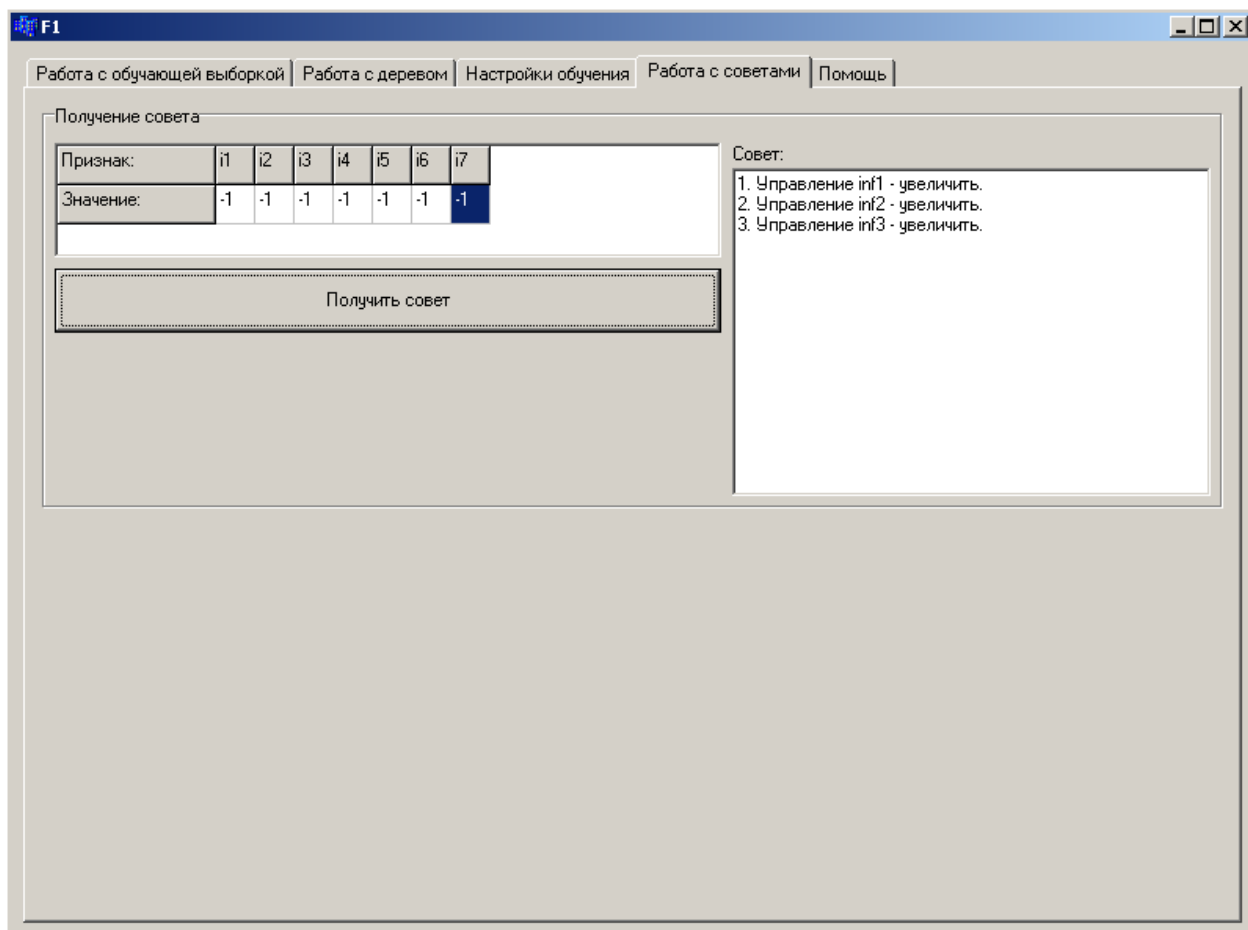


Рисунок 7 – Окно работы с советами

3. Порядок выполнения работы

- 1) Ознакомиться с программой для ситуационного моделирования *Operator24Final.exe*
- 2) Получить у преподавателя вариант задания.
- 3) Настроить программу для выполнения индивидуального задания.
- 4) Подготовить обучающую выборку и занести выборку в программу *Operator24Final*.
- 5) Провести тестирование правильности внесения выборки.
- 6) Построить дерево классификации ситуаций.
- 7) Отобразить графически построенное дерево.
- 8) Проверить работу модели в режиме получения совета для двух ситуаций:
 - задав ситуацию из обучающей выборки;
 - сгенерировав ситуацию, не содержащуюся в обучаемой выборке либо ту, которая не распознана деревом классификации:
- 9) Ввести добавочную ситуацию для обучения, провести настройку и обучение ситуационной модели.
- 10) Повторно проверить работу модели в режиме получения совета.
- 11) Построить дерево классификации ситуаций для скорректированной модели.

12) Оформить отчёт по выполненной работе

4. Содержание отчёта по лабораторной работе

1) Название, цель работы, номер варианта, задание.

2) Настройки программы для выполнения индивидуального задания.

3) Дерево классификации ситуаций для исходных данных в машинном представлении и в графической форме.

4) Результаты проверки работы модели в режиме получения совета для двух ситуаций:

- из обучающей выборки;

- ситуации, не содержащейся в обучающей выборке либо той, которая не распознана деревом классификации.

5) Дерево классификации ситуаций для скорректированной модели.

6) Результаты проверки работы скорректированной модели.

7) Выводы по результатам исследований.

5. Контрольные вопросы

1) Опишите общие сведения о ситуационном моделировании и ситуационном управлении.

2) Алгоритм построения ситуационной модели с помощью процедуры *CLS-9*.

3) Поясните построение дерева классификации при помощи процедуры *CLS-9*.

4) Охарактеризуйте параметры закона управления процессами в СМО.

5) Какие режимы работы имеет моделирующая программа *Operator24Final* и их назначение?

6) Из каких этапов состоит работа программы в режиме обучения?

7) Из каких этапов состоит работа программы в режиме получения совета?

8) Как проводится переобучение модели при отсутствии решения в некоторых ситуациях (в листьях дерева занесен знак «?» - неопределенности)?

6. Список литературы

1. Теория систем и системный анализ в управлении организациями: Справочник: Учеб. пособие/Под ред. В.Н. Волковой и А.А. Емельянова. - М.: Финансы и статистика, 2006. - 848 с.: ISBN 5-279-02933-5.

2. Макаров Р.И., Хорошева Е.Р., Лукашин С.А. Автоматизация производства листового стекла (флоат-способ)/Под ред. Р.И. Макарова; Владим. гос. ун-т. Владимир, 2000. -248с. ISBN 5-89368-206-8.

3. Хант Э., Марин Дж., Стоун Ф. Моделирование процесса формирования понятий на вычислительной машине (пер. с англ.). - М.: Мир, 1970. - 301 с.

4. Информационные технологии в управлении качеством автомобильного стекла: учеб. пособие / Р.И. Макаров [и др.]; Владим. гос. ун-т.-Владимир: Изд-во Владим. гос. ун-та, 2010.-276с. ISBN 978-5-9984-0038-4.

7. Варианты индивидуальных заданий

Вариант 1. Построить ситуационную модель (дерево классификации ситуаций) управляющего воздействия μ_z – расхода природного газа на горелки. Для построения модели использовать выборку, приведенную в таблице 1.

Вариант 2. Построить ситуационную модель (дерево классификации ситуаций) управляющего воздействия $\mu_{ш}$ – положение шиберов дымовой трубы (тяга дымовых газов). Для построения модели использовать выборку, приведенную в таблице 1.

Вариант 3. Дана выборка, описывающая процесс формования ленты стекла в флоат-ванне (таблица 2) [1].

Таблица 2- Выборка для разработки ситуационной модели управления процессом формования ленты стекла

Ситуация	Признаки			Управляющие воздействия				
	z	r_1	r_2	t_1	t_{12}	t_{20}	t_{out}	$vbfm$
1	-1	0	0	+	+	=	=	-
2	0	+1	0	=	+	-	-	=
3	0	0	+1	=	+	-	+	=
4	-1	+1	0	+	+	-	-	-
5	0	+1	+1	=	+	-	=	=
6	-1	0	+1	+	+	-	+	-
7	-1	+1	+1	+	+	-	=	-

Обозначения.

Признаки: z -оптические искажения типа «зебра»; r_1 - оптические искажения типа «растр 1»; r_2 - оптические искажения типа «растр 2»;

Управляющие воздействия: t_1 - температура в первом пролете флоат-ванны; t_{12} - температура в 12-м пролете флоат-ванны; t_{20} - температура в 20-м пролете флоат-ванны; t_{out} - температура ленты стекла на выходе флоат-ванны; $vbfm$ -скорость бортоформирующих машин.

Построить ситуационную модель (дерево классификации ситуаций) управляющего воздействия t_1 - температура в первом пролете флоат-ванны. Для построения модели использовать выборку, приведенную в таблице 2.

Вариант 4. Построить ситуационную модель (дерево классификации ситуаций) управляющего воздействия t_{12} - температура в 12-м пролете флоат-ванны. Для построения модели использовать выборку, приведенную в таблице 2.

Вариант 5. Построить ситуационную модель (дерево классификации ситуаций) управляющего воздействия t_{20} - температура в 20-м пролете флоат-ванны. Для построения модели использовать выборку, приведенную в таблице 2.

Вариант 6. Построить ситуационную модель (дерево классификации ситуаций) управляющего воздействия t_{out} - температура ленты стекла на выходе флоат-ванны. Для построения модели использовать выборку, приведенную в таблице 2.

Вариант 7. Построить ситуационную модель (дерево классификации ситуаций) управляющего воздействия v_{bfm} - скорость бортоформирующих машин флоат-ванны. Для построения модели использовать выборку, приведенную в таблице 2.

Содержание

1. Лабораторная работа № 1. Моделирование процесса обслуживания заявок в информационной системе	3
2. Лабораторная работа №2. Построение нечетких моделей.....	14
3 .Лабораторная работа №3. Построение моделей на нейронных сетях.....	29
4 . Ситуационное моделирование в системах управления с активным элементом (человеком)	47
Содержание.....	59