

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ
ФЕДЕРАЦИИ**

Федеральное государственное бюджетное образовательное учреждение
высшего образования

**«ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ
УПРАВЛЕНИЯ
И РАДИОЭЛЕКТРОНИКИ» (ТУСУР)**

**Методические указания к лабораторным работам
по дисциплине “Математическое моделирование”**

Уровень основной образовательной программы: **магистратура**

Направление подготовки магистра: **09.04.04 «Бизнес-информатика»**

Форма обучения: **очная**

Факультет систем управления (ФСУ)

Кафедра автоматизации обработки информации (АОИ)

Курс 1 Семестр 2

Разработчики:
профессор каф. АОИ
_____ Н.В. Замятин

СОДЕРЖАНИЕ

Введение.....	2
1. Лабораторная работа 1. Модель дуополии на рынке интеллектуальных продуктов.....	2
2. Лабораторная работа 2. Модель распространения инноваций.....	12
3. Лабораторная работа 3. Моделирования рыночного равновесия на рынке ИКТ.....	28
4. Лабораторная работа 4. Математическое моделирование вложения инвестиций.....	33
5. Лабораторная работа 5. Нечеткая аппроксимация модели.....	54
6. Лабораторная работа 6. Нейросетевая аппроксимация модели.....	65
7. Список рекомендованной литературы.....	77

1. ВВЕДЕНИЕ

Задачей математического моделирования как научного направления является воссоздание с помощью компьютера представление и исследование моделей различных предметных областей. Из всего многообразия научных и технических исследований, определяемых математическим моделированием, в учебной дисциплине «Математическое моделирование» выбраны направления, связанные с проблемами аналитического, имитационного моделирования а также принципами визуального моделирования и агентного моделирования, а также моделирования бизнес-процессов.

Лабораторная работа №1

Математическая модель дуополии на рынке интеллектуальных продуктов.

Цель работы. Изучить способы определения эффективной цены на программный продукт в ситуации конкуренции на рынке ИКТ коммерческого и некоммерческого разработчиков программного продукта.

Теоретические сведения.

Статистический анализ рынка серверных операционных систем показывает, что в настоящее время происходит жесткая конкуренция коммерческого продукта Windows и некоммерческого продукта Linux. Операционные системы Windows и Linux занимают самые крупные ниши рынка, оставляя на долю MacOS лишь весьма незначительный рынок графических рабочих станций, при этом доля UNIX_систем неуклонно уменьшается.

Сравнение операционных систем Windows и Linux.

Программное обеспечение делится на 2 большие категории: платное ПО и бесплатное ПО. Платное программное обеспечение подразумевает под собой плату за лицензию на его использование.

Самым известным и широко используемым платным ПО является операционная система Windows. Также платным является использование популярных программ пакета Microsoft Office – Word, Excel, Outlook и др.

Альтернативой платному ПО является свободное программное обеспечение. Самой известной свободной операционной системой является GNU/Linux.

Linux отдали предпочтение такие компании, как Intel, IBM, Hewlett-Packard, Motorola, Nokia, Oracle, Google, Raiffeisen Bank, Boeing и много других. Они поддерживают разработчиков Linux и программ для нее, вкладывая сотни миллионов долларов в разработку и улучшение этой операционной системы.

Операционная система MS Windows разрабатывается и поддерживается одной единственной компанией - Microsoft corporation. Ей же принадлежат авторские права на этот продукт, и она же взимает плату за использование ОС Windows путем продажи лицензии, и бесплатного использования компания

Microsoft не допускает. В Windows графическая среда пользователя является неотъемлемой частью ОС и Windows в варианте без графической среды (консольный режим) попросту не существует. MS Windows как конечный продукт состоит из собственно ОС и небольшого набора прикладных программ, имеющих скромную функциональность, и для решения каких-либо задач нужно дополнительно устанавливать прикладные программы.

К плюсам ОС MS Windows можно отнести.

- Поддержка большого ассортимента компьютерного оборудования с необходимыми драйверами.
- Большое количество прикладных программ (более ста тысяч наименований).
- Большое количество специалистов, хорошо знающих семейство ОС Windows.

ОС MS Windows, минусы.

- Сравнительно высокая стоимость. Варианты Windows независимые от компьютера имеют цену ближе к 200 долларам США и выше. Это стоимость Windows только для одного компьютера.
- Существует большое количество вредоносных программ (компьютерные вирусы) для воздействия на эту ОС. Для версии Windows XP это особо серьезная проблема, которая вынуждает конечного пользователя нести дополнительные расходы на покупку хорошей антивирусной программы либо на обращение к специалистам в случаях, когда вредоносные программы делают невозможной нормальную работу ОС Windows. Эту проблему можно уменьшить за счет квалифицированной настройки ОС Windows и аккуратного ее использования в ситуациях риска, главная из которых Интернет.
- Жесткая зависимость от разработчика. ОС Windows распространяется только в бинарном виде, который труднодоступен для изменения, но более того, компания Microsoft вообще запрещает вносить какие-либо изменения в рабочие коды ОС Windows.

GNU/Linux это ОС, разрабатываемая и поддерживаемая десятками, если не сотнями компаний в разных странах мира и тысячами программистов. Права на эту ОС переданы в общественную собственность. И хотя в мире есть много компаний, предоставляющих платную техническую поддержку этой ОС, само использование GNU/Linux не оговаривается финансовыми условиями.

В то время как GNU/Linux это ОС текстового (консольного) режима и графическая среда это отдельный программный продукт, подобно текстовому редактору или видеопроигрывателю. Программ, которые реализуют графическую среду для ОС GNU/Linux более 10, но широко используются только две - GNOME и KDE.

GNU/Linux в чистом виде, как ОС, распространяется лишь в узкоспециальных областях применения, а для общего применения используются дистрибутивы, состоящие из ОС GNU/Linux, графической среды пользователя и набора прикладных программ, т.е. существует готовая к практическому использованию система, в которую ничего больше не нужно добавлять.

Linux никому конкретно не принадлежит и существуют разные графические среды и разные прикладные программы то, как следствие этого сотни компаний или даже просто группы частных лиц предлагают десятки дистрибутивов Linux.

ОС GNU/Linux, плюсы.

- Сравнительно низкая стоимость.
- Практическое отсутствие, по крайней мере, на сегодняшний день, вредоносных программ для этой платформы.
- Независимость от разработчика. Если требуется какая-то функциональность, отсутствующая в ОС Linux, ее можно добавить, потому что ОС Linux распространяется не только в бинарном виде, но и в исходных кодах, причем нет никаких запретов на модификацию этих исходных кодов.

ОС GNU/Linux, минусы.

- Значительно худшая, чем для платформы Windows, поддержка компьютерного оборудования, в особенности внешнего, такого как, например принтеры или USB устройства.
- Значительно меньшее, чем для платформы Windows, количество прикладных программ. Под ОС Linux пока нет соответствующих версий этих программ, и сопоставимых по функциональности программ. К таким прикладным программам относятся продукты компании Adobe, экономические программы 1С, программа инженерного проектирования AutoCAD, программы распознавания текстов (FineReader). Однако под ОС Linux есть и графические редакторы и программы моделирования/проектирования, но они сильно уступают другим лидерам ОС.
- Меньшее, чем для платформы Windows, количество специалистов.

Как видно плюсы и минусы этих двух операционных систем противоположны, поэтому эти составляющие рынка операционных систем целесообразно выбрать в качестве исследования в лабораторной работе. Стоимость внедрения некоммерческой серверной операционной системы Linux и аналогичной коммерческой Windows в предположении, что стоимость технического обслуживания (администрирования) для двух платформ одинакова, Поэтому стоимость владения серверной операционной системой Linux ниже по сравнению с Windows. При использовании Linux затраты на оплату труда системных администраторов, как правило, выше, но ненамного (для модельной организации разница составляет около 10 000 долл. США в год) [2].

Суммарные расходы на программное обеспечение при внедрении операционной системы Linux состоят примерно из 100 долл., которые тратятся на серверную лицензию. Суммарные расходы на программное обеспечение

при внедрении операционной системы Windows складываются из стоимости лицензий на серверную операционную систему клиентских лицензий.

Установка одного сервера под управлением Linux обходится в 6000 долл. за сам сервер и 250 долл. за его установку. Установка четырех серверов под управлением Windows обойдется в 25 000 долл. В предположении, что стоимость технического обслуживания (администрирования) для двух платформ одинакова, приходим к очевидному выводу о существенно более низкой стоимости владения серверной операционной системой Linux по сравнению с Windows.

Практика показывает, что программные продукты, распространяемые на условиях открытого лицензионного соглашения GNU (например, операционная система Linux и Web-сервер Apache), превосходят своих коммерческих конкурентов (соответственно, Windows и Internet Information Server), помимо стоимости владения, по крайней мере, еще по двум параметрам:

- □ количеству дефектов;
- □ скорости реакции на сообщения пользователей о найденных дефектах.

Существует высокая конкуренция коммерческого и некоммерческого программного обеспечения на рынке серверных продуктов (серверных операционных систем, Web-серверов и т. п., где коммерческие и свободные продукты делят рынок приблизительно поровну), поскольку пользователи этих продуктов — системные администраторы и профессиональные программисты — способны полноценно использовать возможности изучения открытого кода и его модификации.

На рынке клиентских продуктов пользователи, как правило, не ощущают преимуществ от использования открытого кода, поскольку, не обладая квалификацией разработчика, невозможно ни разобраться в «устройстве» продукта, ни модифицировать его под свои нужды (так, например, число инсталляций свободного офисного пакета OpenOffice не сравнимо с числом инсталляций его коммерческого аналога — Microsoft Office).

В качестве коммерческого продукта выберем операционную систему Microsoft Windows а его некоммерческий аналог — Linux.

Microsoft, производитель операционной системы Windows, стремится максимизировать свою прибыль, устанавливая цену лицензии на использование своего продукта, в отличие от партнерства разработчиков операционной системы Linux, распространяемой свободно (т. е. бесплатно и с возможностью изменения исходных кодов на условиях copyleft).

Допущения:

- изначально пользователь ориентирован на использование коммерческого продукта (Windows), и только его высокая цена может заставить пользователя приобрести альтернативный некоммерческий продукт (Linux).
- переменные издержки будем считать нулевыми (действительно, стоимость изготовления копии программного продукта на компакт-диске или ее размещения в интернете пренебрежимо мала по сравнению с затратами на проектирование и разработку).
- каждый пользователь приобретает один и только один продукт: или Windows, или Linux (т. е. одновременная установка двух операционных систем не практикуется — на рынке серверных операционных систем данное предположение вполне соответствует действительности).
- распространением пиратских копий в данной модели пренебрежем.

Обозначения

C — цена лицензии на право использования серверной операционной системы Windows;

q_{\max} — емкость рынка;

$x_W(c)$ — количество пользователей Windows при цене лицензии, равной c ден. ед.;

$x_L(c)$ — количество пользователей Linux при такой цене лицензии Windows;

d — постоянные издержки, которые Microsoft относит на производство серверных операционных систем Windows.

$\Pi_W(c)$ - прибыль коммерческого производителя Microsoft от про

Примем, что функция спроса на Windows линейна:

$$x_W(c) = q_{\max} - bc$$

где $b > 0$, а все пользователи на рынке, которые не приобрели Windows, бесплатно устанавливают Linux:

$$x_L(c) = q_{\max} - x_W(c) = bc$$

Необходимо максимизировать прибыль коммерческого производителя в конкуренции с некоммерческим. Задача линейного программирования, которая стоит перед корпорацией Microsoft — это максимизация прибыли

$$\Pi_W(c) = cx_W(c) - d = c(q_{\max} - bc) - d \rightarrow \max$$

путем установления цены лицензии C , потому что оптимальная цена лицензии доставляет производителю коммерческого продукта наибольшую прибыль в ситуации конкуренции с некоммерческим продуктом.

Оптимальная цена лицензии

$$c^* = \frac{q_{\max}}{2b}$$

соответствует точке максимума квадратичной функции

$$\Pi_W(c) = c(q_{\max} - bc) - d = -bc^2 + q_{\max}c - d$$

При такой цене лицензии Windows спрос на эту операционную систему составит

$$x_W^* = x_W(c^*) = q_{\max} - bc^* = \frac{q_{\max}}{2}$$

при этом спрос на Linux будет равен

$$x_L^* = x_L(c^*) = q_{\max} - x_W(c^*) = \frac{q_{\max}}{2}$$

а прибыль Microsoft —

$$\Pi_W^* = \Pi_W(c^*) = c^* x_W(c^*) - d = \frac{q_{\max}^2}{4b} - d$$

Задание на лабораторную работу

1. Построить линейные зависимости спроса на операционную систему от цены на лицензию для коммерческого $x_W(c)$ и некоммерческих $x_L(c)$ производителей и найти равновесную точку.
2. Построить зависимость прибыли для коммерческого производителя от емкости рынка и определить эффективную стоимость лицензии.
3. Вывести зависимость максимизации прибыли от продажи лицензии для коммерческого производителя.
4. Доказать, что оптимальная цена лицензии будет равна $c^* = \frac{q_{\max}}{2b}$ и рынок будет делиться пополам, воспользовавшись материалом [2]
5. Сформулировать выводы относительно выгод конкуренции коммерческих и некоммерческих производителей для потребителей операционных систем и производителей.
6. расчеты и построения зависимостей выполнить в пакете excel (методика работы приведен в приложении)

Варианты на лабораторную работу

Номер варианта	Цена лицензии С	Издержки d	Объем рынка q_{\max}
1.	100	100000	1000
2.	110	100000	1000
3.	120	100000	1000
4.	130	100000	1000

5.	140	100000	1000
6.	150	100000	1000
7.	160	100000	1000
8.	170	100000	1000
9.	180	100000	1000
10.	190	100000	1000
11.	1000	1000000	2000
12.	1100	1000000	2000
13.	1200	1000000	2000
14.	1300	1000000	2000
15.	1400	1000000	2000
16.	1500	1000000	2000
17.	1600	1000000	2000
18.	1700	1000000	2000
19.	1800	1000000	2000
20.	1900	1000000	2000

Выводы из лабораторной работы

Потребителям операционных систем конкуренция коммерческого и некоммерческого производителей менее выгодна, чем конкуренция двух производителей, оба из которых максимизируют свою прибыль: равновесному состоянию в смешанной дуополии производителей коммерческого и некоммерческого программного обеспечения соответствует такая же цена $\max q/(2b)$.

Производителю же коммерческого программного обеспечения также предпочтительнее конкурировать с некоммерческим производителем, чем с участником рынка, максимизирующим прибыль.

Контрольные вопросы

1. В чем заключаются плюсы владения ОС Windows?
2. В чем заключаются недостатки владения ОС linux?
3. К каким видам рынков относится рынок ИКТ?
4. Что такое объем рынка?
5. Почему переменные издержки при производстве программных продуктов пренебрежимо малы?

Содержание отчета

1. Цель работы
2. Краткие теоретические сведения
3. Вывод выражения для оптимальной стоимости лицензии
4. Зависимости значений спроса для коммерческой и некоммерческой операционных систем
5. Зависимости прибыли от стоимости лицензии

2. Лабораторная работа 2

Мультиагентная модель диффузии инноваций

Цель работы Изучить методологию моделирования диффузии (распространения) инноваций). Приобрести практические навыки работы с системой AnyLogic при построении агентных моделей распространения диффузий инноваций в виде программного продукта.

Задание. В данной лабораторной работе предлагается изучить агентный подход моделирования сложных систем в виде рынка информационно-коммуникационных технологий. В ходе работы нужно создать классическую модель распространения (диффузии) инноваций и ее расширения, которые демонстрируют возможности AnyLogic для создания агентных моделей.

Теоретические сведения.

За последние полвека накоплен достаточно большой опыт экономико-математического моделирования процесса распространения инновационных продуктов на рынке информационно-коммуникационных технологий.

Схема движения потребителей инновационного продукта между сегментами рынка ИКТ, представлена на рис. 1, на котором

$T(t)$ — суммарное число индивидов на рынке в момент времени t ;

$M(t)$ — суммарное число потенциальных потребителей инновационного продукта на рынке в момент времени t ;

$N(t)$ — суммарное число действующих потребителей инновационного продукта на рынке в момент времени t ;

$m(t) = \frac{dM(t)}{dt}$ - число индивидов, переходящих за бесконечно малый промежуток времени dt с неохваченного рынка на потенциальный;

$n(t) = \frac{dN(t)}{dt}$ - скорость распространения инновации — число индивидов, переходящих за бесконечно малый промежуток времени dt с потенциального рынка на охваченный/ В теории инноваций под диффузией инноваций понимается решение $N = N(t)$ задачи Коши для дифференциального уравнения

$$\frac{dN}{dt} = f(t, N(t))$$

с начальным условием $N(0) = N_0$, где

t — время;

$N(t)$ — объем распространения инновации к моменту t (который определяется обычно количеством проданных экземпляров или количеством действующих потребителей инновационного продукта);

$F(t, N(t))$ — функция, определяющая формулой диффузионной кривой и отражающая определенные предположения о природе процесса распространения инновации, Рис.1.

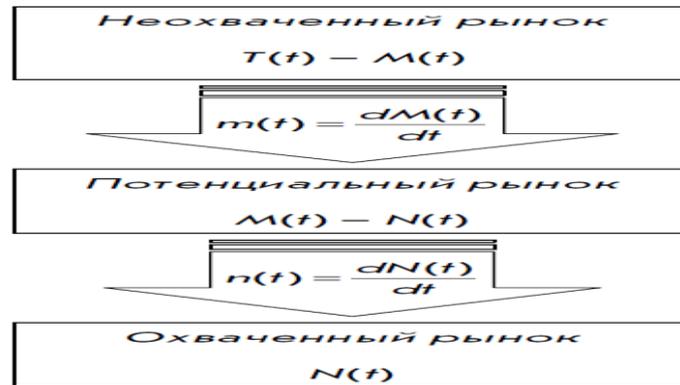


Рис.1 Перемещение потребителей между сегментами инновационного рынка ИКТ

При этом предполагается обычно, что функция $N(t)$ непрерывна и дифференцируема при всех неотрицательных t , а функция $f(t, N(t))$ унимодальна.

Базовая модель диффузии инноваций представляется следующим образом

$$\frac{dN(t)}{dt} = g(t, N(t))(M - N(t))$$

В этой модели общее число потенциальных потребителей инновации M неизменно во времени, а скорость распространения инновации $dN(t)/dt$ в каждый момент времени пропорциональна объему потенциального рынка $M - N(t)$.

По мере увеличения общего числа действующих потребителей инновационного продукта $N(t)$ и соответственно уменьшения числа потенциальных потребителей $M - N(t)$ скорость распространения инновации снижается.

Функция $g(t, N(t))$ в модели (1.3.2) называется в теории инноваций скоростью адаптации, обычно интерпретируется как вероятность того, что потенциальный потребитель инновационного продукта приобретет его в момент t , и считается линейной функцией $N(t)$:

$$g(t, N(t)) = a + bN(t)$$

Подстановка скорости адаптации в базовую модель диффузии инноваций дает следующее обыкновенное дифференциальное уравнение фундаментальной модели диффузии инноваций [2]:

$$\frac{dN}{dt} = (a + bN)(M - N)$$

Параметры a и b в фундаментальной модели диффузии инноваций отражают соответственно степень внешних и внутренних воздействий на скорость адаптации и, следовательно, на скорость распространения инновации.

Внешние влияния на скорость адаптации определяются потребностью пользователей в инновациях и уровнем маркетинговых и рекламных коммуникаций, чему соответствует слагаемое $a(M-N)$ в правой части.

Внутренние влияния на скорость адаптации обусловлены коммуникациями между действующими пользователями инновации и потенциальными потребителями (в результате которых потенциальным потребителям передается информация об инновационном продукте), и этому соответствует слагаемое $bN(t)(M-N)$.

В этом лабораторной работе нужно исследовать влияние внешних воздействий в виде рекламы на продвижение инноваций, (программного продукта) при $b = 0$, тогда :

$$\frac{dN}{dt} = a(M - N)$$

Разделяя переменные в уравнении, имеем

$$\int \frac{dN}{M - N} = \int a dt$$

решение

$$\ln(M - N) = -at + \text{Const}$$

или
$$N(t) = M - e^{-at + \text{Const}}$$

Постоянная интегрирования определяется из начального условия, и окончательно рост охвата рынка во времени описывается функцией

$N(t) = M - (M - N_0)e^{-at}$, графическое изображение которой представлено на рис. 2

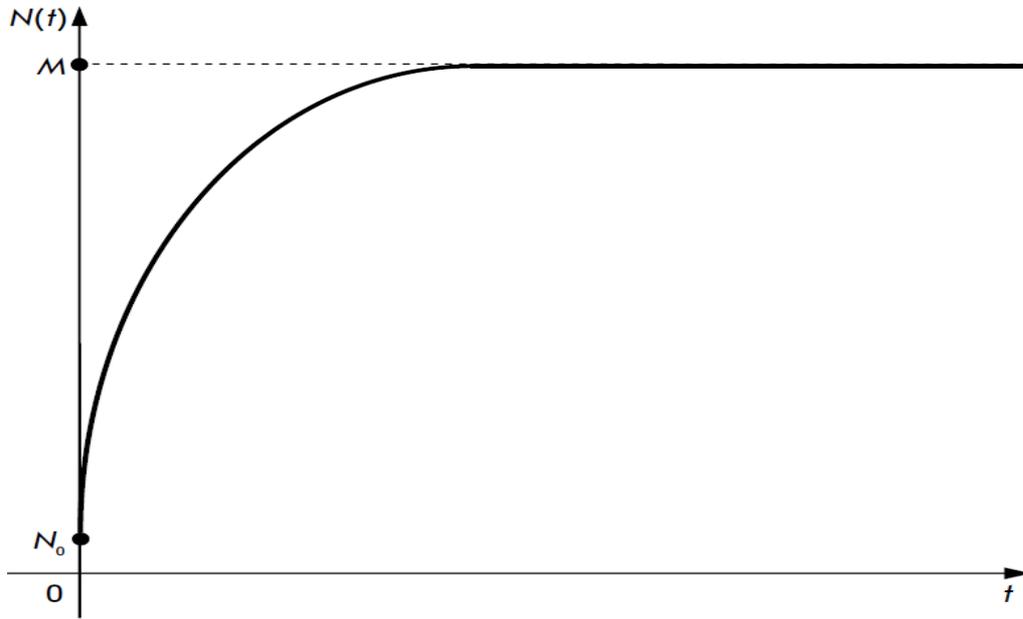


Рис. 2. Рост объема охваченного рынка в моделях внешнего влияния

Агенты в AnyLogic Агент – это некоторая сущность, обладающая активностью, автономным поведением, принимающая решения в соответствии с некоторым набором правил, взаимодействующая с окружением и другими агентами, а также сама при этом изменяющаяся. При помощи агентов целесообразно моделировать рынки ИКТ (агенты – производитель программного продукта и потенциальный пользователь). Активный объект имеет параметры, которые можно изменять извне, переменные, которые можно считать памятью агента, а также поведение (рис. 3).

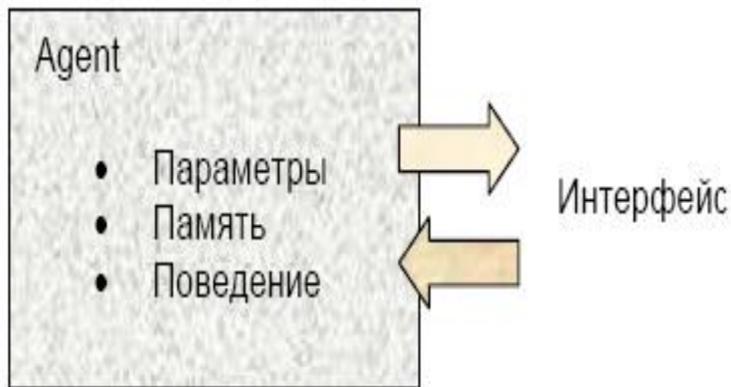


Рис. 3. Вид агента

Ход работы. Создайте проект для своей модели и сохраните его в папке. Первым шагом при создании агентной модели является создание агентов. Для каждого агента задается набор правил, согласно которым он взаимодействует с другими агентами; это взаимодействие и определяет общее поведение системы. В данной работе агентами будут пользователи программного продукта. Создадим агентную модель с помощью Мастера создания модели.

Шаг 1. Щелкните мышью по кнопке панели инструментов Создать. Появится - Agent • Параметры • Память • Поведение Интерфейс. Появится диалоговое окно Новая модель. Задайте имя новой модели. Щелкните мышью по кнопке Далее.

Шаг 2. Выберите шаблон модели (рис. 4). В связи с тем, что создаем новую агентную модель, то нужно установить флажок Использовать шаблон модели и выбрать Агентная модель в расположенном ниже списке Выберите метод моделирования. Щелкните мышью по кнопке Далее.

Шаг 3. Создайте агентов, т.е. задайте имя класса агента и количество агентов, которое будет изначально создано в нашей модели. Задайте в качестве имени класса Person и введите в поле Начальное количество пользователей программного продукта (агентов) 1000. Щелкните мышью по кнопке Далее.

Шаг 4. Задайте свойства пространства, в котором будут обитать агенты и выберите фигуру анимации агента. Установите флажок Добавить пространство и выберите ниже тип этого пространства – Непрерывное. Задайте размер данного пространства: введите в поле Ширина – 600, а в поле Высота – 350. В

результате агенты будут располагаться каким-то образом в пределах непрерывного пространства, отображаемого на презентации модели областью размером 600×350 пикселей. Не меняйте значения, выбранные в выпадающих списках Начальное расположение и Анимация;

Пусть агенты изначально расставляются по пространству случайным образом, а анимируются с помощью фигурки человечка. Щелкните мышью по кнопке Далее

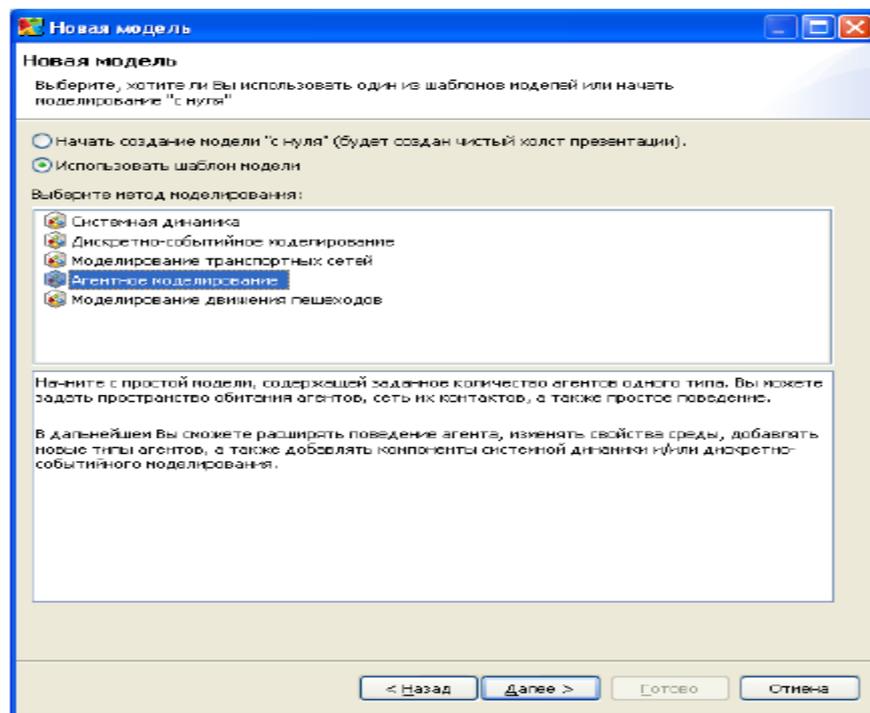


Рис. 4. Шаблон модели

Шаг 4. Задайте свойства пространства, в котором будут обитать агенты и выберите фигуру анимации агента. Установите флажок Добавить пространство и выберите ниже тип этого пространства – Непрерывное. Задайте размер данного пространства: введите в поле Ширина – 600, а в поле Высота – 350. В результате агенты будут располагаться каким-то образом в пределах непрерывного пространства, отображаемого на презентации модели областью размером 600×350 пикселей. Не меняйте значения, выбранные в выпадающих списках Начальное расположение и Анимация;

Пусть агенты изначально расставляются по пространству случайным образом, а анимируются с помощью фигурки человечка. Щелкните мышью по кнопке Далее.

Шаг 5. Задайте сеть взаимосвязей агентов (рис. 5). Установите флажок Использовать сеть и оставьте выбранной опцию Случайное. Ниже можно установить флажок Показывать связи, чтобы отображать на презентации связи между знакомыми (или потенциально могущими встретиться и пообщаться) агентами с помощью линий. Щелкните мышью по кнопке Далее.

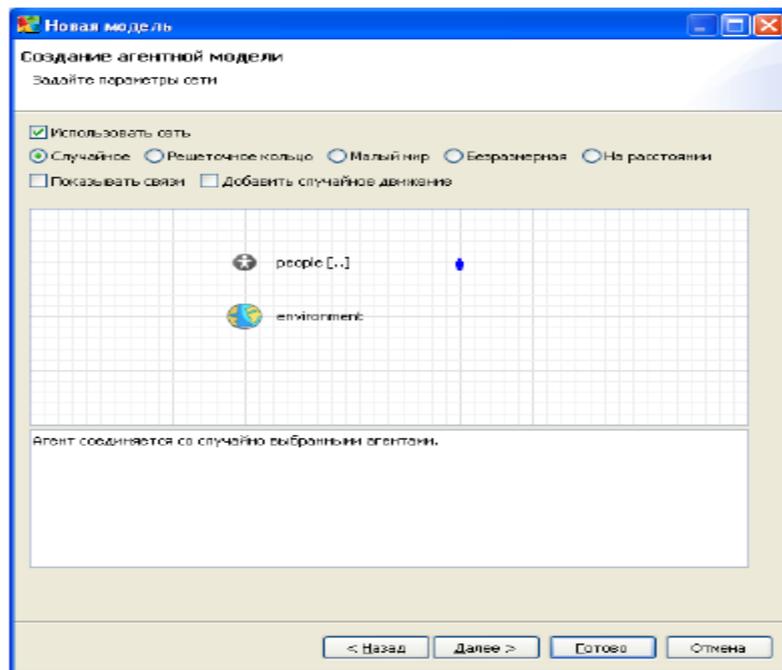


Рис. 3. Сеть взаимосвязи агентов

Шаг 6. Установите флажок Добавить простое поведение. В результате у агента будет создана диаграмма состояний.

Задание характеристик агента Характеристики агента задаются с помощью параметров класса. Все агенты обладают общей структурой, поскольку все они задаются объектами одного класса. Параметры же позволяют задавать характеристики индивидуально для каждого агента.

Создадим параметр, задающий подверженность пользователя влиянию рекламы. Откройте структурную диаграмму класса Person. Перетащите элемент Параметр из палитры Основная на диаграмму класса, в окне свойств параметра задайте имя AdEffectiveness, значение по умолчанию – 0.011.

Задание поведения агента Поведение агента обычно описывается в классе этого агента (в этой модели - класс Person) с помощью диаграммы состояний (стейтчарт). Мастер создания моделей уже создал простейшую диаграмму состояний из двух состояний, между которыми существует два разнонаправленных перехода. Изменим данный стейтчарт.

1. Откройте структурную диаграмму класса Person. На диаграмме класса видно следующую диаграмму состояний (рис. 6).

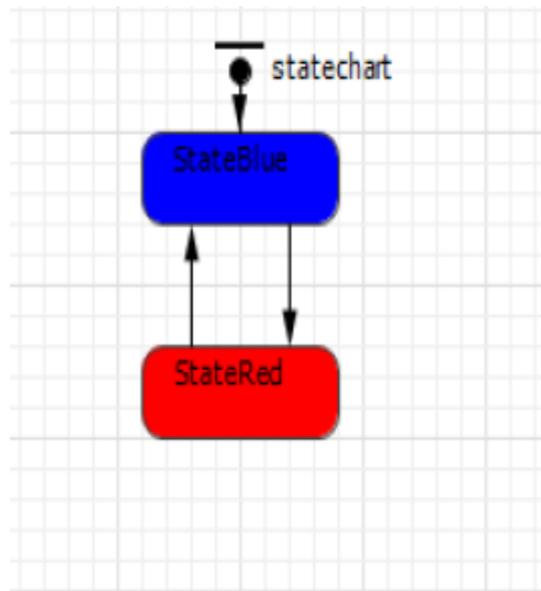


Рис. 6. Диаграмма состояний

2. Откройте свойства верхнего состояния, переименуйте верхнее состояние в PotentialAdopter. Это начальное состояние. Нахождение стейтчарта в данном состоянии означает, что пользователь еще не купил программный продукт.

3. Нижнее состояние назовите Adopter (т.е. пользователь уже купил программный продукт).

4. Измените свойства перехода из состояния PotentialAdopter в состояние Adopter. Этот переход будет моделировать покупку программного продукта. В окне свойств перехода выберите C заданной интенсивностью из выпадающего списка Происходит и введите AdEffectiveness в расположенном ниже поле Интенсивность. Время, через которое пользователь купит программный продукт, экспоненциально зависит от эффективности рекламы p программного продукта.

5. Удалите переход, ведущий из нижнего состояния в верхнее, поскольку пока создается простейшая модель, в которой пользователь, однажды приобретший продукт, навсегда остается его потребителем, и соответственно обратного перехода из состояния Adopter в состояние PotentialAdopter пока что быть не должно (рис. 7). Чтобы удалить переход, выделите его на диаграмме и нажмите Del.

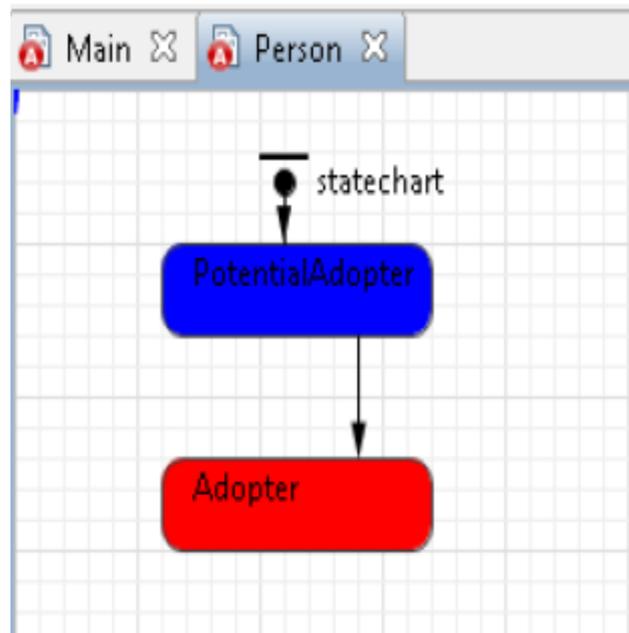


Рис. 7. Стейчарт

6. Настройте выполнение модели (рис. 8). В окне свойств эксперимента перейдите на вкладку Модельное время и задайте останов модели после 8 единиц модельного времени.

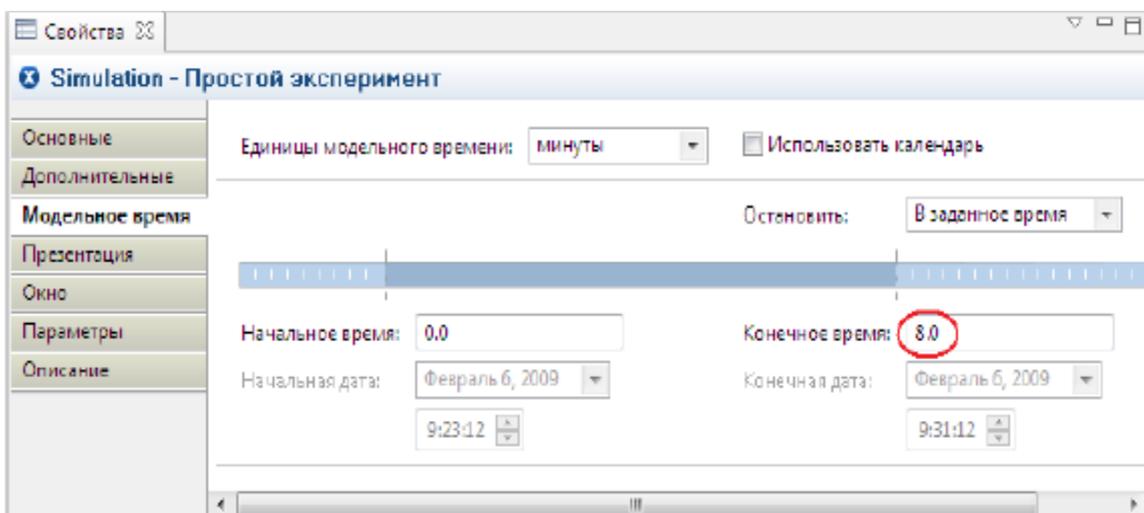


Рис. 8. Установка модельного времени

7. Постройте проект с помощью кнопки панели инструментов Построить (клавиша F7). Если ошибок в проекте нет, то запустите модель. Вы увидите, как число потенциальных покупателей (синих) переходит в разряд покупателей (красных).

Подсчет потребителей программного продукта Главная задача модели распространения инноваций– изучение того, как быстро пользователи покупают новый программный продукт. Для этого будем подсчитывать число пользователей и потенциальных пользователей программного продукта, что можно сделать с помощью функций сбора статистики. Создадим функции сбора статистики для подсчета потенциальных пользователей программного продукта.

1. Откройте диаграмму класса Main. Выделите на диаграмме вложенный объект people.
2. Перейдите на вкладку Статистика панели свойств объекта people. Щелкните мышью по кнопке Добавить функцию сбора статистики. Откроется секция свойств для задания свойств новой функции сбора статистики по элементам этого реплицированного объекта (people).
3. Задайте имя функции – potentialAdopters. Оставьте выбранный по умолчанию Тип функции – количество. Задайте Условие: `item.statechart.isStateActive(item.PotentialAdopter)` Эта функция будет вести подсчет количества агентов, для которых выполняется заданное условие, т.е. тех агентов, которые находятся в текущий момент времени в состоянии PotentialAdopter (являются потенциальными потребителями программного продукта). Здесь item – это агент (элемент реплицированного объекта people).
4. Создайте еще одну функцию сбора статистики (рис. 9). Назовите ее adopters. Тип функции – количество. Условие: `item.statechart.isStateActive(item.Adopter)` Данная функция будет вести подсчет количества агентов, которые находятся в состоянии Adopter (т.е. уже приобрели программный продукт).

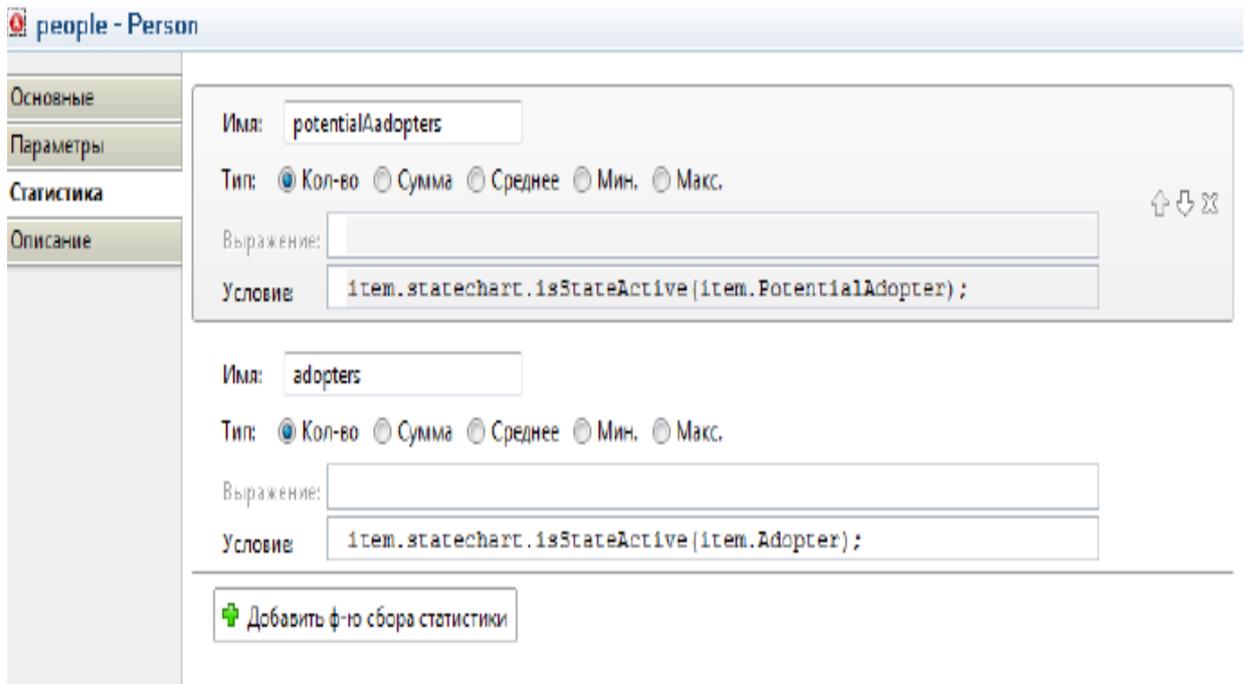


Рис. 9. Функция сбора статистики

Добавьте временной график, отображающий динамику изменения численностей потребителей и потенциальных потребителей продукта. Сконфигурируйте выполнение модели, для этого необходимо настроить текущий эксперимент модели. Если сейчас запустить модель, то она будет работать бесконечно. Поскольку нужно наблюдать поведение модели только тогда, когда происходит процесс распространения информационного продукта, нужно остановить модель, когда система придет в точку равновесия. Процесс распространения информационного продукта в этой модели длится примерно 5 лет. Поэтому задайте останов модели после 5 единиц модельного времени.

Для этого в окне свойств эксперимента *Simulation:Main* перейдите на вкладку **Модельное время**, выберите **В заданное время** из выпадающего списка **Остановить**. В расположенном ниже поле введите 5. Модель остановится после того, как истекнут 5 единиц модельного времени. Задайте выполнение модели в режиме реального времени (вкладка **Презентация** окна свойств эксперимента). Задайте скорость выполнения – 2. Можно сменить метод, используемый для решения системы дифференциальных уравнений. Если не указать никакого конкретного метода, т. е. оставить выбранный по умолчанию

метод Automatic, то во время работы модели AnyLogic будет автоматически выбирать численный метод в соответствии с поведением системы. На вкладке **Дополнительные** окна свойств эксперимента выберите метод RK4 из выпадающего списка **Дифф. уравнения**.

Расположите его, как показано на рис. 10.

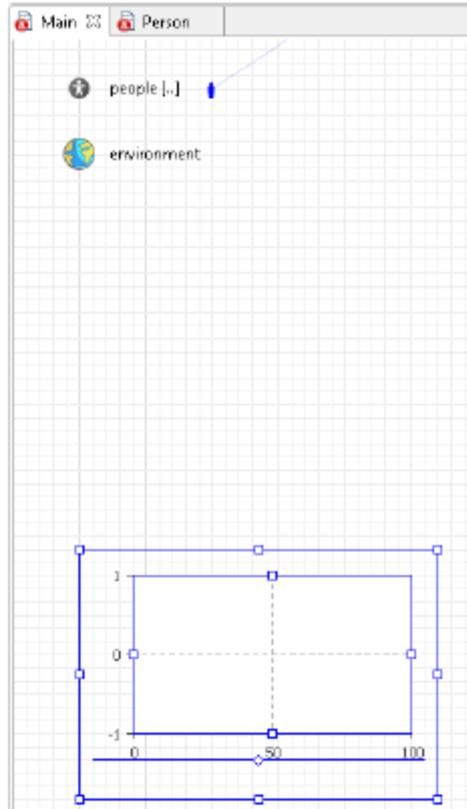


Рис. 10. Выбор метода решения

Настройте свойства графика (рис. 11).

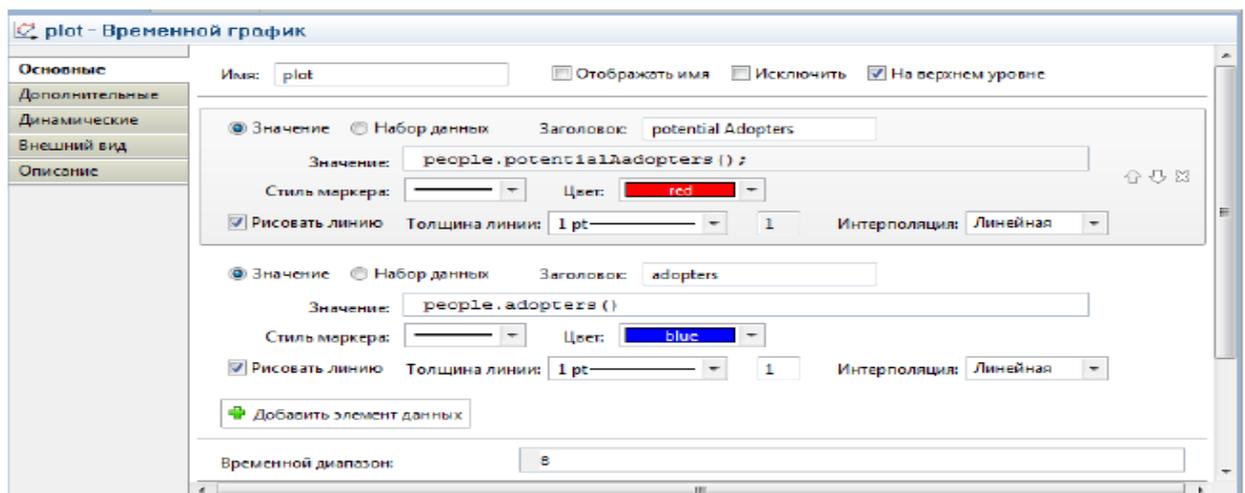


Рис. 11. Настройка свойств графика

Проверьте ошибки и запустите модель. Для проверки ошибок постройте проект с помощью кнопки панели инструментов **Построить** (или клавиша **F7**). В окне **Ошибки** появится список всех ошибок, обнаруженных в проекте, если таковые имеются. Двойным щелчком мыши по ошибке в этом списке можно перейти к предполагаемому месту ошибки, чтобы исправить ее. После построения проекта снова запустите модель.

Просмотрите значения переменных в окне работающей модели. Исследуйте динамику обеих составляющих потока продаж. Для этого откройте окно инспекта для переменной *AdoptionFromAd* в окне презентации. Можно переключить окно инспекта в режим графика – оно будет отображать временной график изменений значения переменной в модельном времени. Текущее значение переменной будет отображаться рядом с началом координат графика. Окно инспекта автоматически масштабируется таким образом, чтобы полностью вместить кривые графиков от начала до конца периода моделирования. Откройте окно инспекта переменной *AdoptionFromWOM* и переключите его в режим графика (рис. 12).

Запустите модель. На графике (рис. 12) проследите динамику моделируемого процесса. Под влиянием рекламы каждую единицу времени постоянная доля от общей численности потенциальных потребителей программного продукта приобретает распространяемый программный продукт.

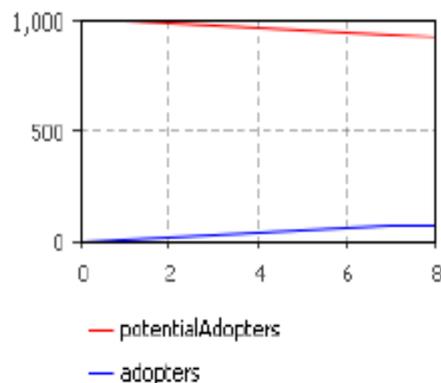


Рис. 12. График процесса продаж

При внедрении нового информационного продукта на рынок, когда число потребителей равно нулю, реклама будет являться единственным источником

продаж. Наибольший рекламный эффект отмечается в начале процесса распространения информационного продукта; он неуклонно падает по мере уменьшения численности потенциальных потребителей

Варианты заданий

Вариант	Эффективнос- ть рекламы	Длительность процесса
1	0,010	6
2	0,010	7
3	0,010	8
4	0,005	9
5	0,005	10
6	0,005	11
7	0,015	10
8	0,015	9
9	0,015	8
10	0,010	7
11	0,010	6
12	0,010	7
13	0,005	8
14	0,005	9
15	0,005	10
16	0,015	11
17	0,015	10
18	0,015	9
19	0,020	8
20	0,020	7

Контрольные вопросы

1. В чем суть диффузии инноваций?
2. Что такое внутренние и внешние воздействия на диффузию инноваций?

3. К какому дифференциальному уравнению сводится диффузия инноваций?
4. Что такое мультиагентные системы?
5. Что такое стейтчарт?

Содержание отчета

1. Цель работы
2. Кратко основные теоретические сведения
3. Диффузия инноваций в виде ОДУ Коши
4. Скриншоты работы модели диффузии инноваций
5. Графики диффузии инноваций в виде программного продукта
6. Ответы на контрольные вопросы

4. Лабораторная работа 3

«Моделирования рыночного равновесия на конкурентном рынке ИКТ»

Цель: Построить компьютерную модель влияния спроса и предложения на динамику цен рыночного равновесия.

Теоретическая часть

Разработчики или посредники поставляют на рынок информационный продукт. Чем больше рыночная цена, тем больше поставщиков и товара. Потребители покупают товар. Чем меньше цена, тем больше покупателей и покупок. Товар на рынке характеризуется двумя параметрами: количеством и ценой.

Задание. Для изучения выбирается очень грубая математическая модель: линейная, без запасов, случайностей, прогнозов и прочих факторов.

Функция зависимости спроса от цены

$$D_{md} = D_0 - K_d * P_{rc},$$

где – D_{md} - спрос (demand) за текущий интервал времени;

D_0 – спрос при нулевой цене; K_d – крутизна линии спроса; P_{rc} – цена (price) товара.

Линия зависимости предложения от цены

$$Spl = S_0 + K_s * Prc,$$

где – Spl - предложение (supply) за текущий интервал времени; S_0 – предложение при нулевой цене; K_s – крутизна линии спроса; Prc – цена (price) товара.

Ход лабораторной работы

Создать файл DmdSplEqu. Mdl, Рис.1.

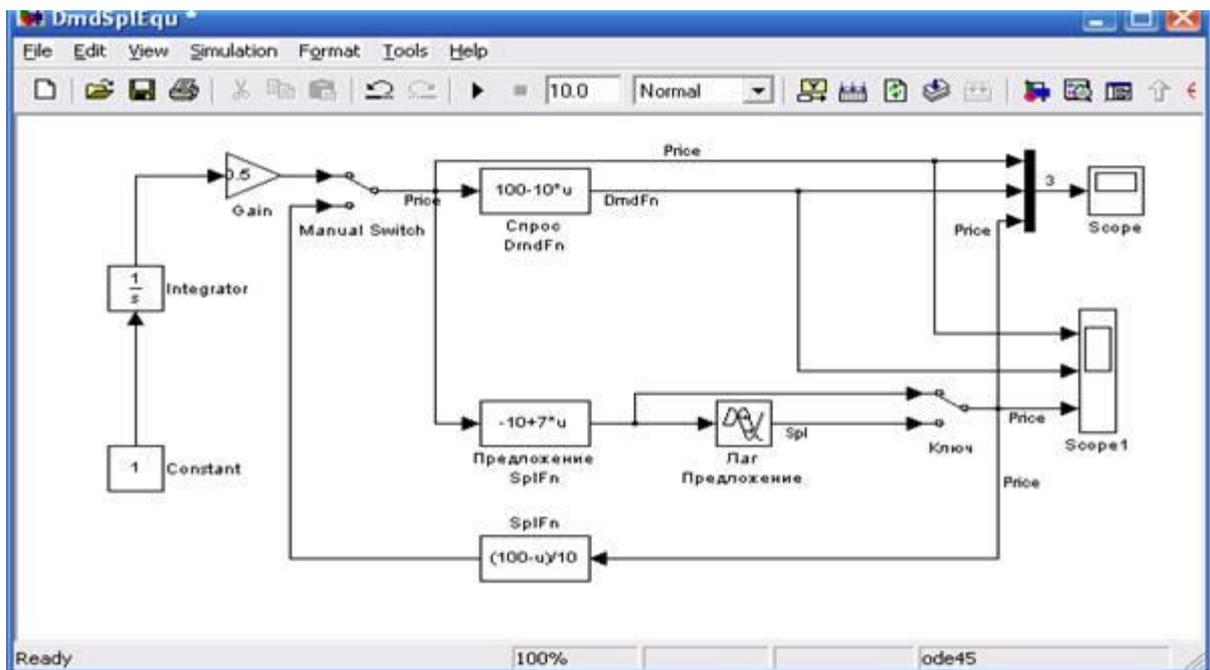


Рис.1. Визуальная модель спроса и предложения

Спрос представлен одним стандартным блоком с именем DmdFn. Он вычисляет значение спроса в зависимости от цены, подаваемой на вход блока. Обозначения и параметры блока на схеме следующие $u = Prc$, $D_0 = 100$, $K_d = 10$.

Предложение представлено тремя стандартными блоками. Собственно функция зависимости количества предлагаемых на продажу товаров от цены реализуется блоком с именем SplFn. Он вычисляет значение предложения в зависимости от цены, подаваемой на вход блока.

Обозначения и параметры блока на схеме следующие:

$$u = Prc, S_0 = 10, K_s = 7.$$

Блок Лаг имитирует запаздывание поставщика на рынке. Продавец поставляет товар в количестве Spl , определенном на основе цен прошлого интервала времени.

Блок $SplFn1$ имитирует решение поставщика смириться с ценой текущего спроса. Он соглашается продать весь товар по цене, которую диктует линия спроса. Блок реализует функцию обратную функции спроса и вычисляет цену Prc , по которой сможет купить весь товар Spl поставщика. Параметры блока одинаковы с параметрами блока Спрос $DmdFn$. Рис.2

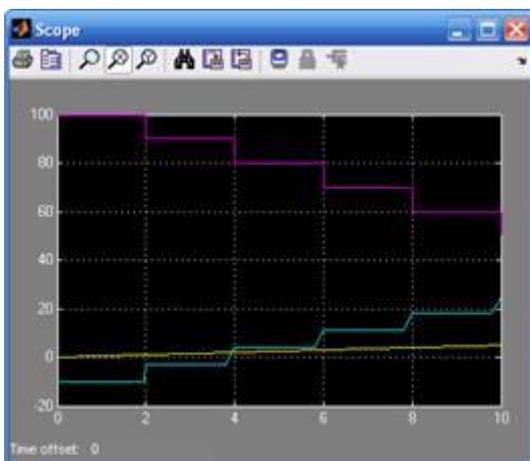
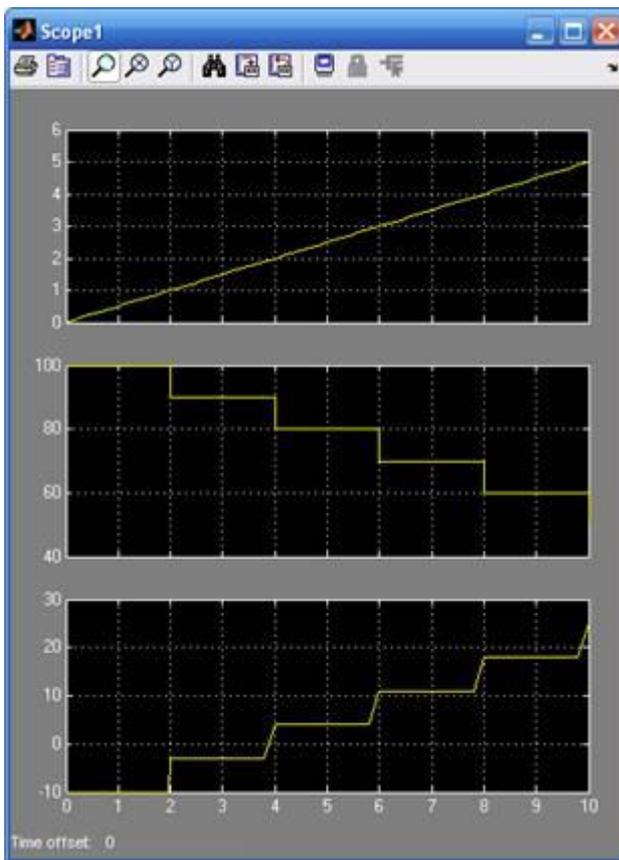


Рис.2. Зависимости цены от спроса и предложения

Внешние средства – программа Matlab.

Автоматизация управления экспериментом осуществляется с помощью программы Matlab

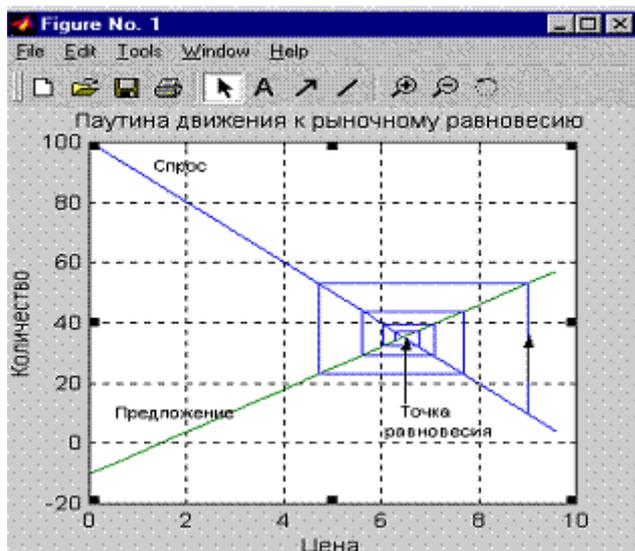


Рис.3. Паутинообразная модель

```
%Market equilibrium price simulation
```

```
% File: C:\Csr_MtLb\DmdSplEquM. m – путь к своей папке
```

```
% 1.Simulate static functions, plot its
```

```
% 2.Simulate price dynamic, plot price Web graphics
```

```
%=====
```

```
% Set model Path – указать путь к своей папке, например
```

```
path(path,'C:\Csr_MtLb\MrktEqLPr')

```

```
%=====
```

```
% 1.Simulate static functions, plot its
```

```
% Load and Run DmdSplEqu. mdl
```

```
open_system('DmdSplEqu')
```

```
sim('DmdSplEqu')% Write Vars into WS from Scope
```

```
% Plot Static features
```

```
plot(ScopeData(:,2),ScopeData(:,3:4))
```

```
hold on
```

grid

pause(5)%Пауза для переключения в режим имитации

%двойным щелчком мыши над ключами

%=====

%2.Simulate price dynamic

sim('DmdSplEqu')

%=====

%3. plot price Web graphics

for i = 2:11 %

Цикл черчения паутины движения к равновесию Рис.3.

line([ScopeData(i-1,2) ScopeData(i,2)],[ScopeData(i,4) ScopeData(i,4)])

line([ScopeData(i,2) ScopeData(i,2)],[ScopeData(i,4) ScopeData(i+1,4)])

end

Варианты заданий

Номер варианта	D0	Kd	. S0	Ks
1.	80	8	10	7
2.	100	10	12	8
3.	120	12	14	9
4.	140	14	16	10
5.	150	16	18	11

Вопросы

1. Что такое точка равновесия на рынках?
2. В чем отличия рынка ИКТ от рынка товаров?
3. Как влияет пиратство на рынках ИКТ?
4. Почему жизненный цикл программных продуктов маленький?

5. В чем особенность транзакционных издержек на рынках ИКТ?

Содержание отчета

- 1 Цель работы
2. Краткие теоретические сведения
3. Краткое описание хода работы
4. Вид визуальной модели спроса и предложения
5. Графики спроса и предложения
6. Рисунок паутины движения к рыночному равновесию
7. Зависимости спроса и предложения на рынках ИКТ
8. Выводы по работе
9. Ответы на контрольные вопросы

Лабораторная работа 4. Математическое моделирование вложения инвестиций

Целью проведения лабораторных занятий является приобретение студентами знаний и практических навыков математического моделирования с использованием пакета прикладных программ Fuzzy Logic Toolbox, входящего в состав программной среды MatLab.

Ход работы. Исходные данные находятся в окне «Эффективность инвестиций» Рис.1. Исходных показателей несколько (5 шт.), и их значения несоизмеримы (представляются в различных шкалах и различных единицах измерения), поэтому для анализа определения качества инвестиционного проекта используется интеллектуальная технология Fuzzy Logic Toolbox.

Вначале студент знакомится с технологией решения задач, руководствуясь настоящими методическими указаниями, в которых детально представлен вариант решения задачи математического моделирования. Затем, для более

глубокого усвоения материала и закрепления навыков работы с ППП, предлагается самостоятельно выполнить задание, вариант которого определяется буквой фамилии студента. Варианты заданий для самостоятельного выполнения приведены в приложении 1.

Результаты самостоятельного решения задач отражаются в отчете. Отчет оформляется в бумажном и электронном варианте, определяемом преподавателем в соответствии с требованиями кафедры и защищается студентом в установленные сроки. Рекомендуется следующая структура отчета:

- титульный лист;
- оглавление;
- введение (1 страница);
- Ход работы
- Ответы на вопросы
- список использованной литературы.

Этап 1. Общая характеристика инвестиционного проекта

Организуется предприятие по разработке сетевых устройств. Планируется разработка сетевых устройств 3 видов: коммутаторы, маршрутизаторы и модемы. Для каждого вида сетевых устройств – отдельная группа разработчиков. Длительность проекта составляет 3 года.

Следует учитывать ряд факторов, существенных для проектов в любой предметной области. К ним относятся:

- общие параметры проекта;
- производственно-технологические особенности;
- внешние финансовые факторы.

К общим параметрам проекта относятся:

- предметная область и номенклатура разрабатываемых сетевых устройств;
- срок проекта;

- стартовый (акционерный) инвестированный капитал;
- процентная ставка желаемой доходности проекта ($r^0\%$).

К группе *производственно-технологических условий* отнесены:

- необходимость помещений для хранения и установки оборудования:
- возможность покупки или аренды помещения;
- цены покупки/аренды помещений;
- характеристики аппаратного обеспечения:
- цена 1 компьютера и сроки поставки;
- цена и длительность работ по вводу аппаратного обеспечения;
- наличие прямых (удельных) затрат на производство:
- структура удельных затрат (на единицу продукции) по номенклатуре продукции: материалы и комплектующие, сдельная зарплата, энергопотребление, иные ресурсы;
- ценовые параметры по видам затрат;
- условно-постоянные издержки:
- штат и затраты на административно-управленческий персонал;
- затраты на содержание зданий.

В группе *внешних финансовых факторов* учитываются:

- цены сбыта производимой продукции с учетом возможности сезонных колебаний,
- система налогообложения предприятия:
- налог на прибыль;
- НДС;
- налог на имущество;
- условия банковского кредитования:
- ставки в зависимости от срока;
- условия обеспечения кредита;
- режим кредитной линии/рамочного кредита;
- условия возврата (помесячно, поквартально, в конце срока);

- ставки рефинансирования и дисконтирования, индекс инфляции.

Этап 2. Описание общих входных данных

Акционер вкладывает деньги в проект общей суммой на 9 000 000 руб. и выходит из проекта по его окончании. Срок инвестиционного проекта по разработке сетевых устройств – 3 года

График производства фиксированный.

Общая ставка дисконтирования для рубля 12%.

Ставка рефинансирования для рубля на 3 года: 15%, 12%, 11%,.

Ставки налогов

НДС, %	Налог на имущество	Налог на прибыль
18	2,2	22

Индекс инфляции

Объект	1-год, %	Тенденции изменения в год %
Сбыт	5	1
Прямые издержки	6	1
Общие издержки	6	1
Зарплата	4	1
Недвижимость	8	1

Математическое моделирование инвестиций на основе нечеткой логики выполняется (Fuzzy Logic) в пакете прикладных программ Fuzzy Logic Toolbox.

Fuzzy Logic Toolbox – пакет расширения программы математического анализа MATLAB, содержащий инструменты для проектирования систем нечеткой логики. Пакет позволяет создавать экспертные системы на основе нечеткой логики, проводить кластеризацию нечеткими алгоритмами, проектировать нечеткие нейронные сети. Базовым понятием *Fuzzy Logic Toolbox* является

FIS-структура (FIS – Fuzzy Inference System) – система нечеткого вывода. Графический интерфейс пакета предназначен для интерактивного пошагового проектирования нечетких систем, функции командной строки используются для разработки программ.

В пакете *Fuzzy Logic Toolbox* реализованы две системы нечеткого вывода: *Mamdani* (типа Мамдани-Заде, задаваемая по умолчанию) и *Sugeno* (типа Такаги-Сугено-Канга). В работе будет использоваться система нечеткого вывода *Mamdani*.

Этап 1. Общая характеристика задачи, решаемой в условиях неопределенности

Применение интеллектуального программного обеспечения, каким является система MATLAB (включающая пакет нечеткой логики Fuzzy Logic Toolbox), позволяет определить качество инвестиционного проекта. Для его применения необходимо указать лингвистические переменные, включаемые в математическую модель, и их терм множества значений.

Входные лингвистические переменные – это название показателей экономической эффективности инвестиционного проекта, являющиеся основой для принятия инвестиционного решения. Следующие показатели будем считать лингвистическими переменными:

- чистая приведенная стоимость (*Net Present Value – NPV*), млн. руб.;
- внутренняя норма рентабельности (доходности) (*Internal Rate of Return – IRR*), %;
- срок окупаемости (*Payback Period – PB*), месяцы;
- средняя норма рентабельности (*Account Rate of Return – ARR*), %;
- индекс прибыльности инвестиций (*Profitability Index – PI*).

Все функции принадлежности типовые и поэтому их можно задавать, пользуясь стандартной библиотекой *Fuzzy Logic Toolbox*. Диапазоны значений лингвистических переменных указываются исходя из значений интегральных фактических показателей,

Показатель	Рубли	Доллар
▶ Ставка дисконтирования, %	12.00	0.00
Период окупаемости - РВ, мес.	24	24
Дисконтированный период окупаемости - DPВ, мес.	27	24
Средняя норма рентабельности - ARR, %	58.92	58.92
Чистый приведенный доход - NPV	5 639 243	324 951
Индекс прибыльности - PI	1.49	1.77
Внутренняя норма рентабельности - IRR, %	47.98	47.98
Модифицированная внутренняя норма рентабельности - MIRR, %	27.44	20.91

Рис.1. Параметры задачи

Первая лингвистическая переменная «*NPV*- чистая приведенная стоимость (*Net Present Value*),» оценивается термами, принимающими значения:

- очень низкая (от 0 до 1,5);
- низкая (от 0,5 до 2,5);
- средняя (от 1,5 до 6);
- высокая (от 5 до 9);
- очень высокая (8 и выше);

Вторая лингвистическая переменная «*IRR* - внутренняя норма рентабельности (доходности) (*Internal Rate of Return*)» оценивается одним термом, принимающим значения:

- приемлемо (выше 12).

Значение ниже является неприемлемым и в нечетком правиле указывается полем «none». Значение 12 отражает ставку дисконтирования из таблицы, приведенной на рис. 78 части1.

Третья лингвистическая переменная «*PВ*- срок окупаемости (*Payback Period*)» оценивается термами, принимающими следующие значения:

- приемлемо (ниже 36).

Значение выше является неприемлемым и в нечетком правиле указывается полем «none». Значение 36 отражает срок окупаемости, указанный на рис. 78 части 1.

Четвертая лингвистическая переменная «ARR - средняя (учетная) норма рентабельности (*Account Rate of Return*)» оценивается термами, принимающими следующие значения:

- низкая (от 0 до 15);
- средняя (от 7 до 30);
- высокая (от 25 до 50);
- очень высокая (40 и выше).

Пятая лингвистическая переменная «PI- индекс прибыльности инвестиций (*Profitability Index*)» оценивается термами, принимающими значения:

- *приемлемый* (выше 1).

Значение ниже является неприемлемым и в нечетком правиле указывается полем «none». Значение 1 отражает следующее требование: индекс не может быть ниже единицы.

Выходная лингвистическая переменная «Качество инвестиционного проекта» – указывает на возможность эффективной реализации проекта, определяется в баллах и может характеризоваться следующими термами:

- очень низкое (от 0 до 20);
- низкое (от 15 до 45);
- среднее (от 45 до 60);
- высокое (от 55 до 85);
- очень высокое (от 80 до 100).

Этап 2. Описание входных и выходных переменных с помощью функций принадлежности

Для анализа полученного варианта инвестиционного проекта создаются нечеткие правила, в основе которых лежат следующие рекомендации:

1. Количество входных переменных в нечеткой модели

Для удобства расчета количество входных переменных модели не должно превышать 7 ± 2 .

2. Количество термов для лингвистической оценки переменных

Минимальное количество термов равно двум, в этом случае переходим к бинарной оценки с нечеткими границами. Максимальное количество термов для лингвистической оценки одной переменной в пределах 5

3. Особенности функций принадлежности крайних термов

При формировании функций принадлежности надо стремиться к тому, чтобы степени принадлежности границ интервала крайним термам были равны единице. В этом случае выполняется естественное правило, заключающееся в том, что чем меньше (больше) значение переменной, тем в большей степени оно соответствует крайнему терму "низкий" ("очень высокий").

Этап 3. Формирование базы правил (модели) системы нечеткого вывода

В основе построения нечетких правил необходимо придерживаться следующей рекомендации:

1. Должно существовать хотя бы одно правило для каждого терма выходной переменной.
2. Для любого терма входной переменной должно существовать хотя бы одно правило, в котором этот терм используется в качестве посылки.
3. Для произвольного вектора входных переменных должно существовать хотя бы одно правило, степень выполнения которого больше нуля. Другими словами, правила базы знаний должны покрывать всю предметную область.

Для примера приведем некоторые нечеткие правила (см. табл. 1).

Таблица 1

Правила нечеткого вывода

	<i>NPV</i>	<i>IRR</i>	<i>PB</i>	<i>ARR</i>	<i>PI</i>		Качество проекта
--	------------	------------	-----------	------------	-----------	--	------------------

ЕСЛ	очень низкая	неприемл емо	неприем лемо	низкое	неприемл емо	ТО	очень низкое
ЕСЛИ	очень высока я	приемле мо	приемле мо	очень высокая	приемле мо	ТО	очень высокое
ЕСЛИ	низкая	неприемл емо	неприем лемо	приемле мая	приемле мо	ТО	среднее
ЕСЛИ	средня я	приемле мо	приемле мо	высокая	приемле мо	ТО	среднее
ЕСЛИ	высока я	приемле мо	приемле мо	очень высокая	приемле мо	ТО	высокое

Этап 4. Технология построения экспертной системы «Проект» средствами Fuzzy Logic Toolbox

1. Запуск FIS-редактора. Для запуска редактора нечеткой системы вывода (Fuzzy Inference System Editor – **FIS Editor**) сначала необходимо выполнить команду **Пуск/Все программы/MATLAB R2011b** или **R2013b**, а затем в появившемся рабочем окне программы MATLAB (рис. 2) набрать **fuzzy** и нажать клавишу ENTER.

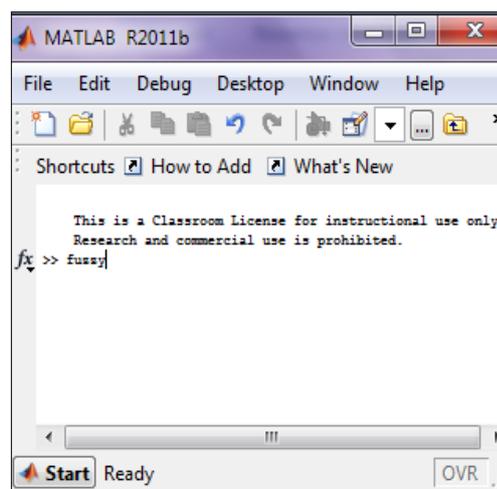


Рис. 2. Рабочее окно программы MATLAB R2011b

Общий вид FIS-редактора представлен на рис. 3.

Главное меню редактора содержит три вкладки.

- **File** – создание, сохранение, считывание и печать файлов;
- New FIS** – выбор типа задаваемой системы нечеткого вывода; типа Mamdani или Sugeno;
- Import** – загрузка файлов: **From Workspas** (из рабочего пространства программы MATLAB) или **From Disk** (из внешнего файла);
- Export** – позволяет сохранить редактируемую систему;
- To Workspas** (в рабочем пространстве системы MATLAB) или **To Disc** (во внешнем файле);
- **Edit** – редактирование и исключение входных и выходных переменных:
- Undo** – отменяет последнее действие;
- Add Variable** – позволяет добавить переменную типов **Input** или **Output**;
- Remove Selected Variable** – удаляет выбранную переменную;
- Membership Functions** – вызывает редактор функций принадлежности;
- Rules** – вызывает редактор правил вывода.

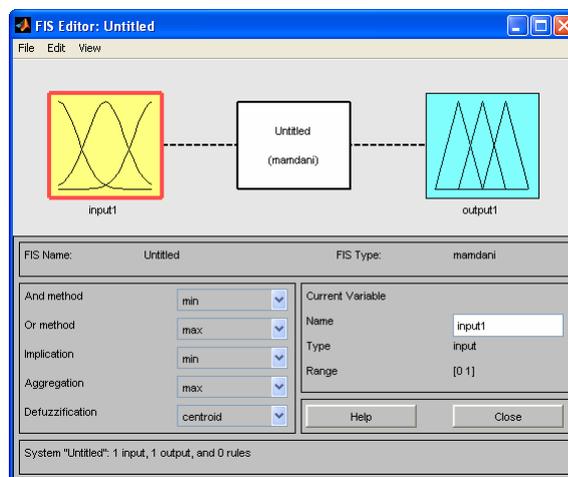


Рис. 3. Общий вид FIS-редактора

- **View** – дополнительные возможности:
 - Rules** – вызывает программу нечеткого вывода с правилами;
 - Surface** – графическое представление результатов расчетов.
- Параметры, располагающиеся в нижней части окна, можно оставить без изменения (по умолчанию). Они имеют следующее содержание.

- **Add metod** – метод логической конъюнкции (метод минимального значения (нечеткое логическое И (min), **prod** – метод алгебраического произведения, **Custom** – метод, определяемый пользователем);
- **Or metod** – метод логической дизъюнкции (метод максимального значения (нечеткое логическое ИЛИ (max), **probor** – метод алгебраической суммы);
- **Implication metod** – метод вывода заключения (**min** – метод минимального значения, **prod** – метод алгебраического произведения);
- **Aggregation metod** – метод агрегирования (**max** – метод максимального значения, **sum** – метод ограниченной суммы, **probor** – метод алгебраической суммы);
- **Defuzzyfication metod** – метод дефаззификации (**centroid** – метод центра тяжести, **bisektor** – метод центра площади, **mom** – метод среднего максимума, **som** – метод наименьшего модального значения; **lom** – метод наибольшего модального значения).

2. Указание количества входов и выходов. Начальное состояние системы характеризуется наличием одного входа и одного выхода. Так как в системе «Проект» пять входов («NPV», «IRR», «PB», «ARR» и «PI») и один выход («Вероятность принятия проекта»), необходимо через пункт меню **Edit/Add variable/input** последовательно добавить еще четыре входа.

3. Идентификация входов и выходов. Для идентификации входов и выходов используется редактор функций принадлежности **Membership Function Editor**. Пункт меню **Edit** (редактирование) содержит следующие операции:

Add Variable... – добавление встроенной функции принадлежности;

Remove Current MF – удаление функции принадлежности;

FIS Properties – вызов **FIS** редактора;

Rules – вызов редактора нечетких правил.

Однократным щелчком левой кнопки мыши по блоку **input1** нужно изменить в *поле имени* его имя на «NPV», а затем сделать щелчок кнопкой мыши в

любом месте экрана. В результате первый вход получает имя «NPV». Те же действия произведите с оставшимися входами и выходом.

4. Сохранение полученного результата. Для сохранения полученного результата воспользуйтесь именем «INV». Сохранение приложения можно выполнить через следующие пункты меню: **File/Export/to File/INV** (рис. 4).

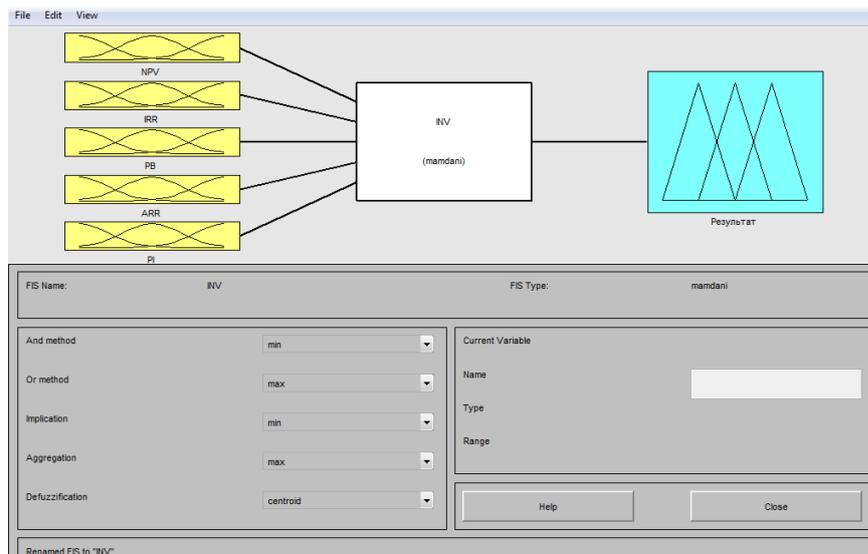


Рис. 3. Присваивание имен входам и выходу и сохранение приложения

5. Вызов окна для задания функций принадлежности. Программу-редактор функций принадлежности **Membership Function Editor** можно открыть двумя способами:

1) **Edit/Membership Functions;**

2) двойным щелчком левой кнопки мыши по соответствующему входу.

6. Задание функций принадлежности. Для переменных, относящихся к непрерывному виду данных, функцию принадлежности удобнее для наглядности изобразить графически. Часто используются кусочно-линейные функции принадлежности: треугольная (*trimf*) и трапециевидная (*trapmf*).

Функция принадлежности, принимающая вид симметричного треугольника, зависит от двух параметров, один из которых определяет максимум данной функции, второй задает ширину основания функции.

Для трапециевидной функции – основание трапеции выражает все допустимое множество значений фактора, верхнее – тех значений, когда эксперт устанавливает гарантированное соответствие этих значений выбранному значению лингвистической переменной. Боковые ребра трапеции отражают изменение степени уверенности эксперта в его классификации от 1 до 0 и обратно.

Пользователь должен подобрать такую функцию принадлежности, которая наилучшим образом отражает его понимание нечеткой зависимости.

Задание функций начнем с переменной «NPV». Для этого необходимо щелкнуть указателем курсора на данном входе, а затем через меню **Edit/Add MFs** перейти к окну **Membership Functions**, с помощью которого задается желаемый вид функции принадлежности (рис. 5). Последовательно задайте трапециевидную, треугольную и еще одну трапециевидную функцию.

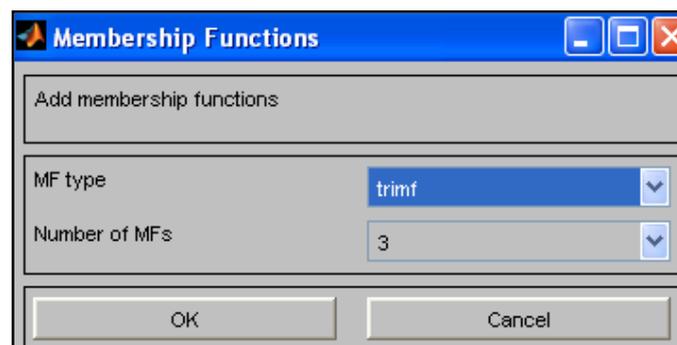


Рис. 4. Задание вида функции принадлежности

После нажатия кнопки **OK** происходит возврат в окно редактора функций принадлежности, в котором уже будут представлены все три заданные функции.

7. Редактирование функции принадлежности. Для редактирования функции принадлежности в окне **Membership Functions Editor** в поле **Rang Display Rang** необходимо установить диапазон изменения переменной «NPV» от 0 до 1500, подтвердив ввод нажатием клавиши **Enter** или щелчком мыши в любом месте экрана. Выделим первый терм (получив красный цвет) в поле **Name**, укажем его значение «низкий» и передвинем кривую таким образом,

чтобы она соответствовала нужным значениям. С помощью поля **Params** (параметры) можно подогнать график более точно (рис. 6).

На рис. 5 показано, что чистая приведенная стоимость оценивается пятью функциями: для значения *очень низкая* и *очень высокая* – трапециевидной, *низкая*, *средняя* и *высокая* – треугольной.

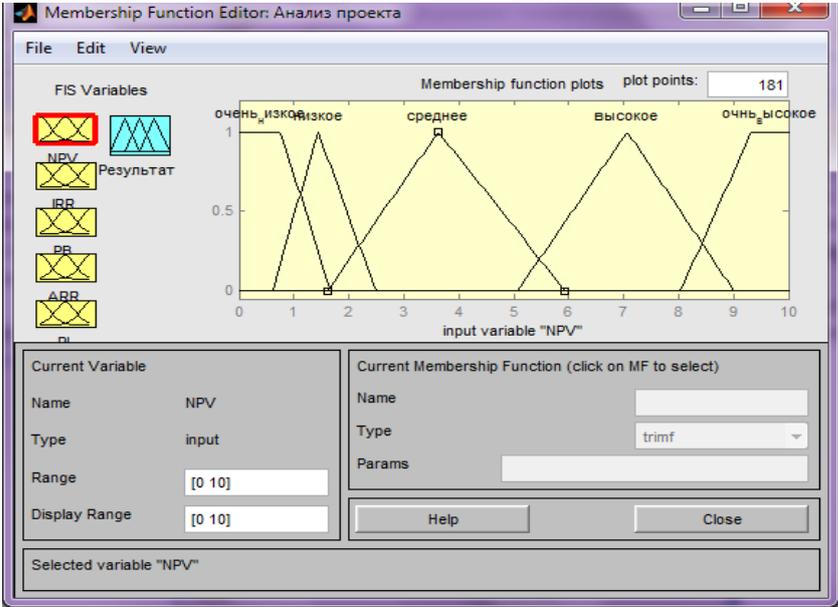


Рис. 6. Функции принадлежности входной переменной (NPV) “Чистая приведенная стоимость”

Аналогично осуществляется редактирование функций принадлежности входных переменных «IRR», «РВ», «ARR», «PI», и выходной переменной «Результат». Итоги данной операции представлены на рис. 7 – 10.

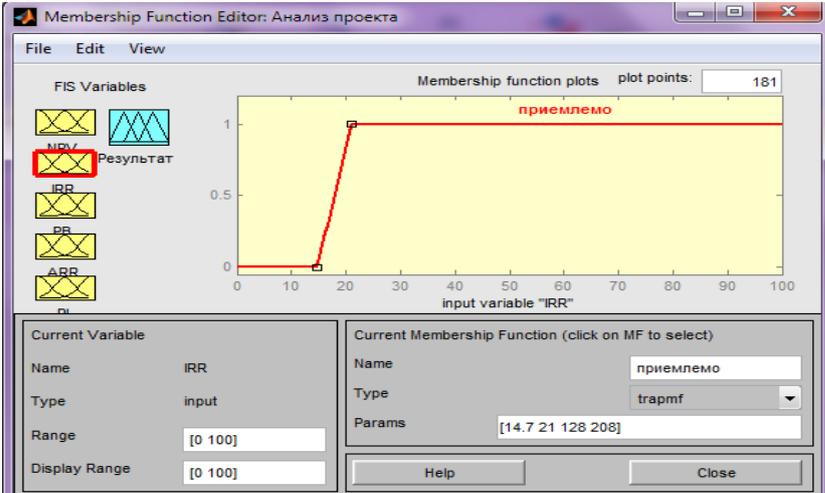


Рис. 7. Функция принадлежности входной переменной (IRR) “Внутренняя норма рентабельности”

Так как IRR не может быть ниже ставки дисконтирования, указанной в проекте (12%), поэтому здесь применяется одна функция принадлежности, отражающая приемлемое значение IRR. Если это значение ниже ставки дисконтирования, то оно не указывается в нечетких правилах вывода.

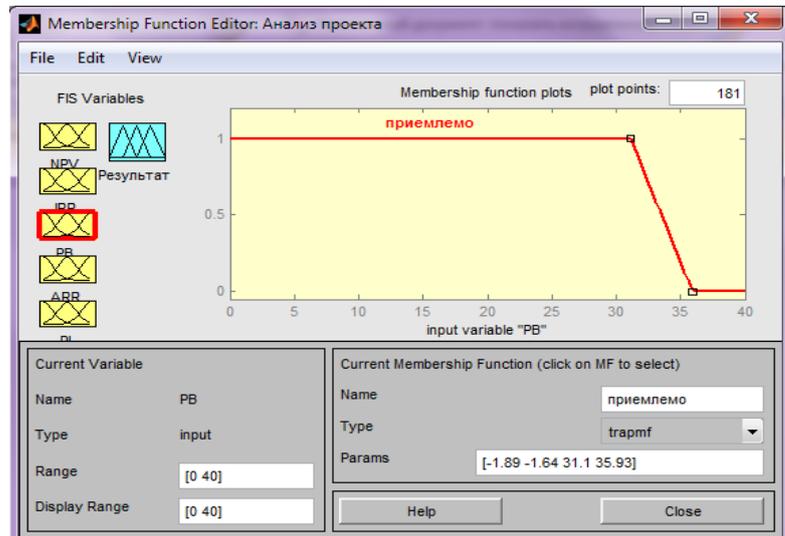


Рис. 8. Функция принадлежности входной переменной (PB) “Срок окупаемости”

Здесь также как и с IRR. Срок окупаемости не может быть ниже указанного в проекте (36 мес), поэтому используется только одна функция принадлежности.

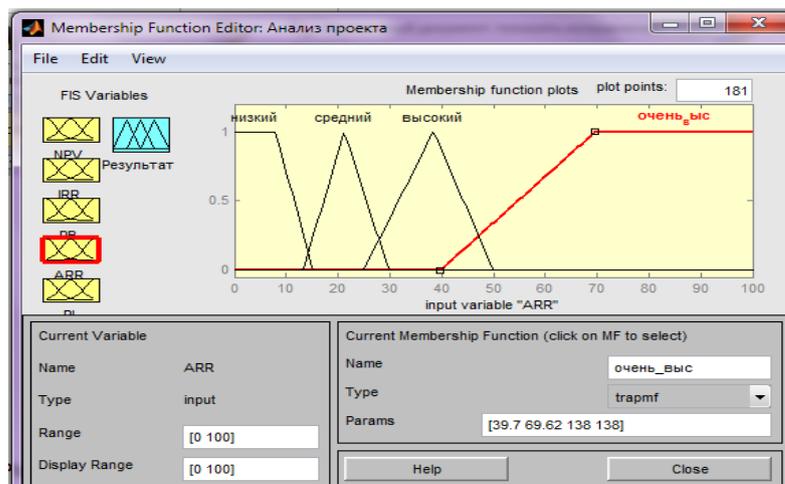


Рис. 9. Функция принадлежности входной переменной (ARR) “Средняя норма рентабельности”

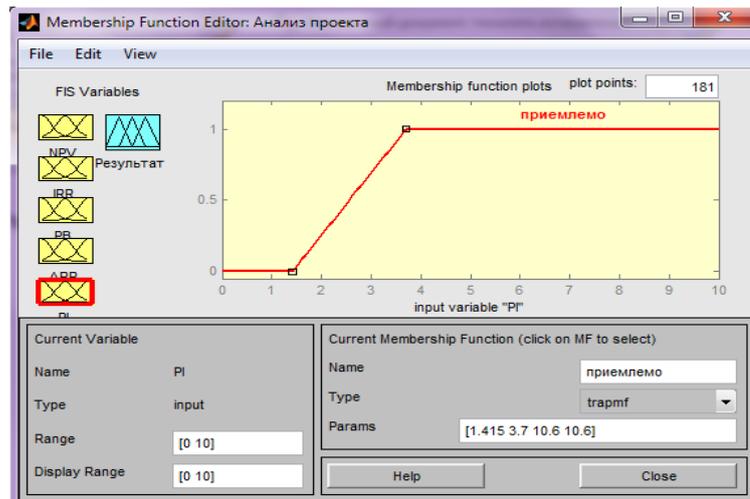


Рис. 10. Функции принадлежности входной переменной (PI) “Индекс прибыльности инвестиций”

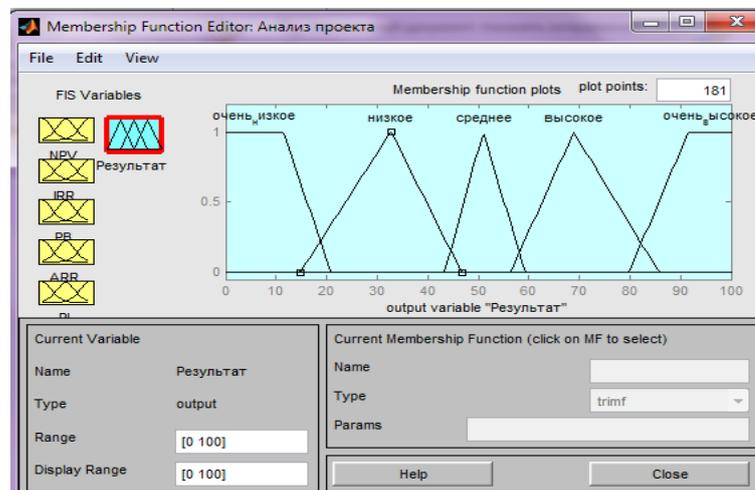


Рис. 11. Функции принадлежности выходной переменной “Качество инвестиций”

Качество проекта оценивается баллами и представляется пятью треугольными функциями.

Для закрытия окна **Membership Function Editor: INV** необходимо нажать кнопку **Close**.

8. Создание и редактирование правил. Для создания и редактирования базы правил систем нечеткого вывода следует воспользоваться пунктами меню **Edit/Rules** или дважды щелкнуть по центральному квадрату INV.

Ввод правил осуществляется в соответствии с табл. 2 указанием соответствующих значений термов (рис. 12) в окне редактора правил системы нечеткого вывода **Rule Editor**.

Добавление правила осуществляется нажатием кнопки **Add rule**, удаление – **Delete rule**, изменение его веса – полем **Weight** (вес), а изменение в выделенном правиле – полем **Change rule**. В качестве связки термов в поле **Connection** указывают либо логическое И (and), либо логическое ИЛИ (or).

9. Применение системы нечеткого вывода для формирования решения.

Для этого необходимо открыть окно просмотра правил нечеткого вывода **View/Rules** и установить исходные значения переменных. Пример оценки эффективности проекта по алгоритму Мамдани представлен на рис. 12.

В центральной части расположены прямоугольники, соответствующие отдельным входным переменным правил вывода.

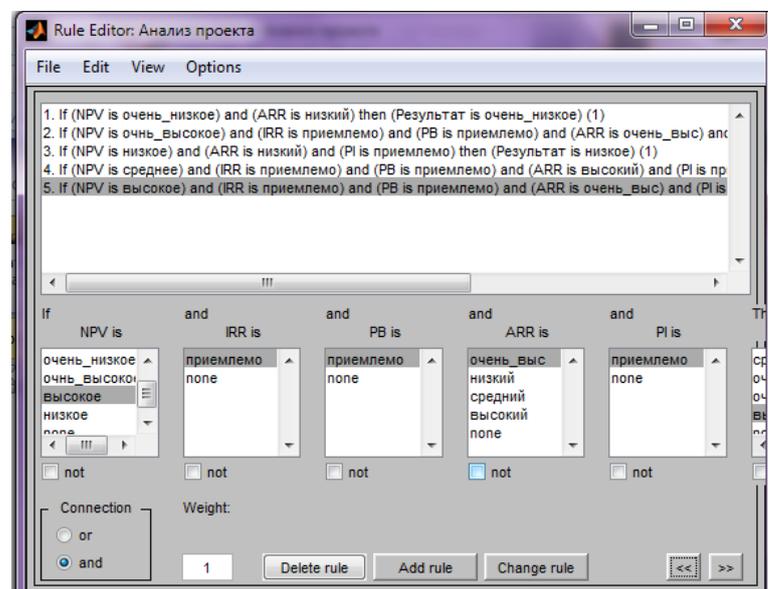


Рис. 12. Создание и редактирование базы правил

Каждому правилу соответствует отдельная строка из этих прямоугольников. В нижней части расположен прямоугольник, отображающий результат дефаззификации выходной переменной после аккумуляции всех заключений правил нечеткого вывода.

Ввод исходных данных осуществляется из окна, приведенного на рис. 13.

Эффективность инвестиций		
Длительность проекта: 36 мес.		
Период расчета: 36 мес.		
	Рубли	Доллар
► Ставка дисконтирования, %	12.00	0.00
Период окупаемости - PB, мес.	24	24
Дисконтированный период окупаемости - DPB, мес.	27	24
Средняя норма рентабельности - ARR, %	58.92	58.92
Чистый приведенный доход - NPV	5 639 243	324 951
Индекс прибыльности - PI	1.49	1.77
Внутренняя норма рентабельности - IRR, %	47.98	47.98
Модифицированная внутренняя норма рентабельности - MIRR, %	27.44	20.91

Рис. 13. Исходные данные для анализа проекта

Ввод осуществляется путем смещения вертикальных красных линий (влево, вправо), так чтобы значения показателей, находящихся в верхней части окна (рис. 14) совпали с рис. 13. В результате в правом столбце (в верхней части) будет отражаться качество проекта в баллах.

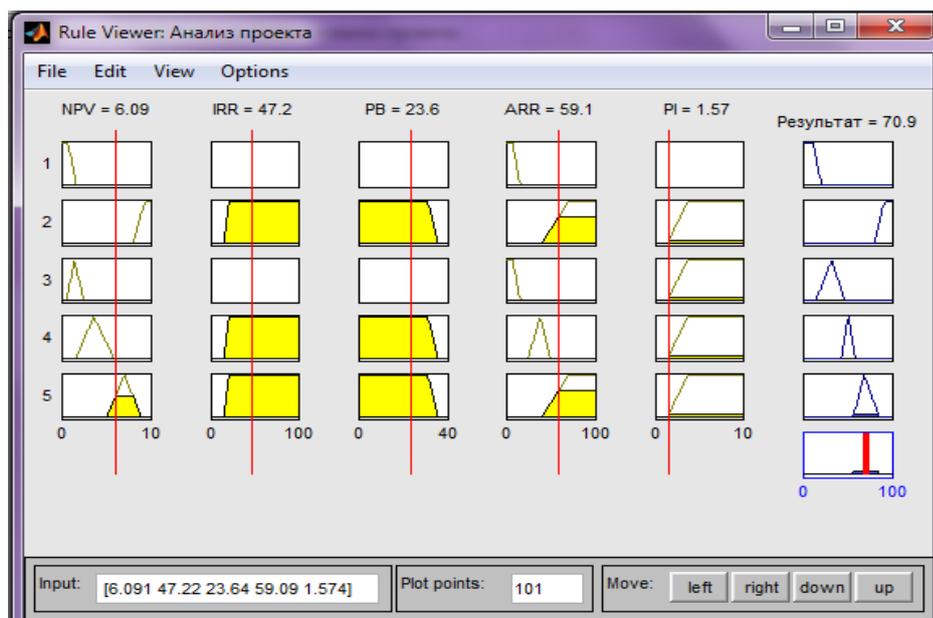


Рис. 14. Итоги работы системы после ввода фактических исходных данных

Этап 5. Анализ полученных результатов

Как видно, из рис. 14, при значениях входных переменных: NPV = 5,6 млн., IRR = 48%, PB = 24 мес., ARR = 59,9, PI = 1,49 качество проекта оценивается 70,9 баллов из ста, что вполне приемлемо.

Изменить текущие исходные данные можно, сдвигая вертикальные красные линии в соответствии с изменившейся ситуацией в левой части окна **Rule Viewer**.

В меню **Options (Сервис)** содержатся следующие операции: **Format** – выбор формата записи правил (**Verbose** – в форме текста, **Symbolic** – в символической форме, **Indexed** – в цифровой форме). В правой части указывается результат нечеткого вывода. Если требуется графическое представление зависимости результата от исходных переменных, то его можно увидеть, воспользовавшись пунктом меню **View/Surface** (просмотр поверхности системы нечеткого вывода).

Этап 6. Оформление отчета

Оформление отчета выполняется по данным методических указаний. В отчете должны быть представлен текст с демонстрацией важных результатов с помощью скриншотов.

Варианты заданий на лабораторную работу

№ варианта	Название	Этапы	Продукция	Материал
1	Издательств о журнала по информацио нным систем ам	Аренда помещения, покупка оборудования, пуск/наладка	Журнал, рекламные объявления, буклеты «Деловой досуг»	Бумага, чернила, картон, переплет, ламинат
2	Компания по производств у средств связи	Покупка здания, оборудования и пуск/наладка	Коммуникационн ое оборудование	Комплектующи е, программное обеспечение
3	Компания по предоставле	Аренда помещения,	Информационные услуги	Столы, компьютеры,

	нию информацио нных услуг	оборудования, пуск/наладка		коммуникацион ное оборудование
4	Издательств о журнала по компьютерн ой тематике	Аренда помещения, покупка оборудования, пуск/наладка	Книги, журналы, буклеты (не менее 3-х видов)	Бумага, чернила, картон, переплет, обложки на календари
5	Издательств о журнала по автомобильн ой тематике	Аренда помещения, покупка оборудования, пуск/наладка	Журнал, буклеты, календари.	Бумага, чернила, картон, переплет, пластиковые обложки на календари
6	Компания по разработке программны х продуктов	Аренда помещения, покупка оборудования, пуск/наладка	Программное обеспечение	Стол ы, компьютеры, коммуникацио нное оборудование
7	Компьютерн ая компания	Аренда помещения, покупка оборудования, пуск/наладка	Компьютеры	Комплектующи е, программное обеспечение
9	Питьевая вода в розлив	Покупка земли, оборудования, пуск/наладка	Вода негазированная, газированная, сладкая	Очистители, пластиковые бутылки разных размеров

10	Полиграфическое предприятие	Покупка здания, оборудования, пуск/наладка	Визитки, буклеты, календари	Картриджи (различного цвета), бумага, картон и т.п.
11	Сувенирно-рекламное производство	Покупка помещения, покупка оборудования, пуск/наладка	Сувениры, памятные подарки, рекламные открытки	Бумага, ламинат, сувениры, краски
12	Типография	Аренда помещения, покупка оборудования, пуск/наладка	Стенды, плакаты, буклеты	Бумага, чернила, картон, переплет

Общие параметры

Отвергаются проекты со сроком окупаемости (PB) более указанного в графе 2, внутренней нормой рентабельности (IRR) менее дисконтной ставки, индексом прибыльности (PI) менее 1 и качеством проекта ниже 60 баллов.

№ варианта	Срок проекта, количество лет	Стартовый акционерный капитал, руб.
1	4	9 000 000
2	5	7 000 000
3	5	10 000 000
4	3	15 000 000
5	5	12 000 000
6	3	10 000 000

7	3	5 000 000
8	4	4 000 000
9	4	6 000 000
10	5	7 000 000
11	4	8 000 000
12	4	8 000 000

4. Лабораторная работа 5

Нечеткая аппроксимация моделей.

1. Цель работы: изучение основных функций пакета Fuzzy Logic Toolbox программной среды MatLab, а также приобретение навыков построения нечеткой системы для аппроксимации моделей.

2. Теоретические сведения

Назначение и возможности пакета Fuzzy Logic Toolbox

Пакет Fuzzy Logic Toolbox (пакет нечеткой логики) – это совокупность прикладных программ, относящихся к теории *размытых* или *нечетких* множеств и позволяющих конструировать так называемые нечеткие экспертные и/или управляющие системы.

Основные возможности пакета:

- построение систем нечеткого вывода (регуляторов, аппроксиматоров зависимостей);
- построение адаптивных нечетких систем (гибридных нейронных сетей);
- интерактивное динамическое моделирование в Simulink. Пакет позволяет работу:
 - в режиме графического интерфейса;
 - в режиме командной строки;
 - с использованием блоков и примеров пакета Simulink.

Графический интерфейс Fuzzy Logic Toolbox

Состав графического интерфейса. В состав программных средств Fuzzy Logic Toolbox входят следующие основные программы, позволяющие работать в режиме графического интерфейса:

- редактор нечеткой системы вывода Fuzzy Inference System Editor (FIS Editor или FIS-редактор) вместе со вспомогательными программами – редактором функций принадлежности (Membership Function Editor), редактором правил (Rule Editor), просмотрщиком правил (Rule Viewer) и просмотрщиком поверхности отклика (Surface Viewer);
- редактор гибридных систем (ANFIS Editor, ANFIS-редактор);
- программа нахождения центров кластеров (программа Clustering – кластеризация).

Набор данных программ предоставляет пользователю максимальные удобства для создания, редактирования и использования различных систем нечеткого вывода.

Построение нечеткой аппроксимирующей системы.

Командой (функцией) **Fuzzy** из режима командной строки запускается основная интерфейсная программа пакета Fuzzy Logic – редактор нечеткой системы вывода (Fuzzy Inference System Editor, FIS Editor, FIS-редактор).

Вид открывающегося при этом окна приведен на рис. 1.

Главное меню редактора содержит позиции:

File – работа с файлами моделей (их создание, сохранение, считывание и печать);

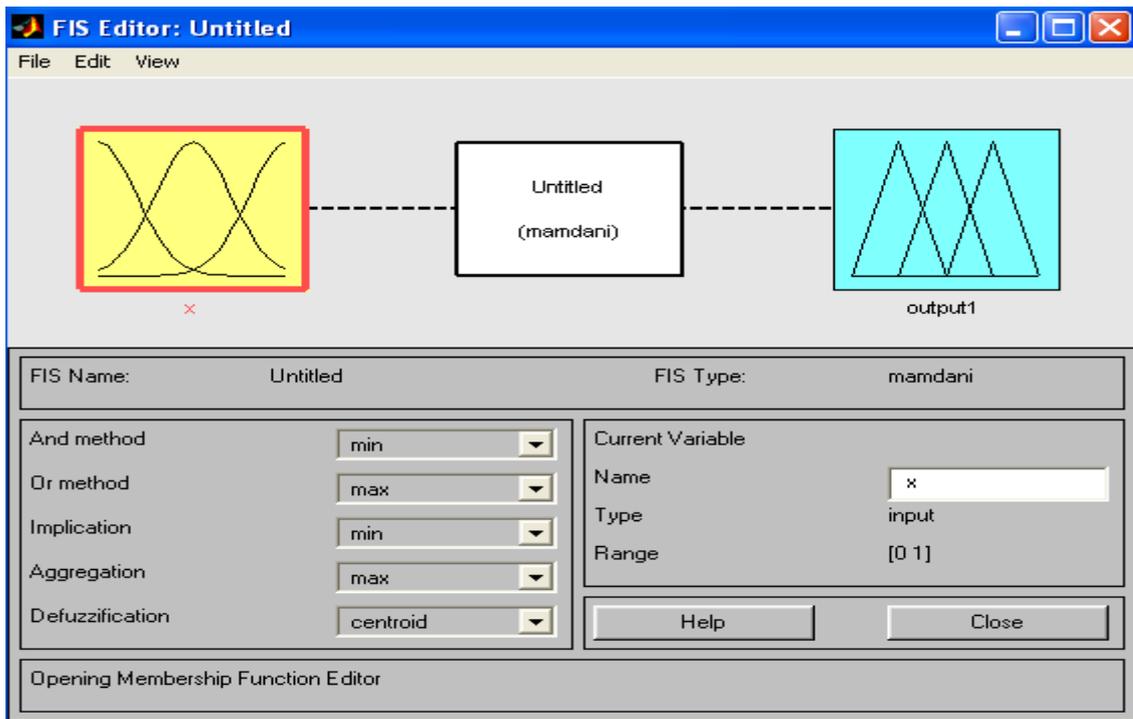


Рис. 1. Вид окна Fis Editor

Edit – операции редактирования (добавление и исключение входных и выходных переменных);

View – переход к дополнительному инструментарию.

Разработаем нечеткую систему, отображающую зависимость между переменными x и y , заданную с помощью табл.1, представленные в таблице данные отражают зависимость $y = x^2$.

Таблица 1. Значения x и y

x	-1	- 0.6	0	0.4	1
y	1	0.36	0	0.16	1

Требуемые действия отобразим следующими пунктами.

1. В позиции меню File выбираем опцию New Sugeno FIS (новая система типа Sugeno), при этом в блоке, отображаемом белым квадратом, в верхней части окна редактора появится надпись Untitled2 (sugeno).

2. Щелкнем левой кнопкой мыши по блоку, озаглавленному input1 (вход 1). Затем в правой части редактора в поле, озаглавленном Name (Имя), вместо input1 введем обозначение нашего аргумента, т.е. x . Обратим внимание, что

если теперь сделать где-нибудь (вне блоков редактора) однократный щелчок мыши, то имя отмеченного блока изменится на x ; то же достигается нажатием после ввода клавиши Enter.

3. Дважды щелкнем по этому блоку. Перед нами откроется окно редактора функций принадлежности – Membership Function Editor (см. рис. 2). Войдем в позицию меню Edit данного редактора и выберем в нем опцию Add MFs (Add Membership Functions – Добавить функций принадлежности). При этом появится диалоговое окно (рис.), позволяющее задать тип (MF type) и количество (Number of MFs) функций принадлежности (в данном случае все относится к входному сигналу, т. е. к переменной x). Выберем гауссовы функции принадлежности (gaussmf), а их количество зададим равным пяти – по числу значений аргумента в табл. 1. Подтвердим ввод информации нажатием кнопки ОК, после чего произойдет возврат к окну редактора функций принадлежности, рис.2.

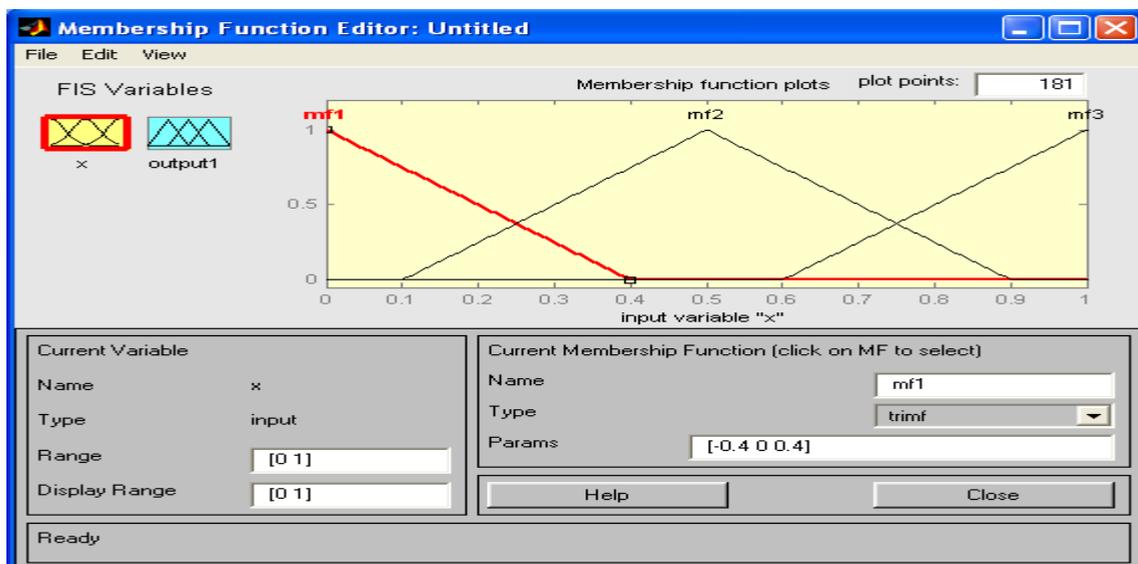


Рис. 2. Окно редактора функций принадлежности

4. В поле Range (Диапазон) установим диапазон изменения x от -1 до +1, т.е. диапазон, соответствующий табл. 1. Щелкнем затем левой кнопкой мыши где-нибудь в поле редактора (или нажмем клавишу ввода Enter). Обратим внимание, что после этого произойдет соответствующее изменение диапазона в поле Display Range (Диапазон дисплея).

5. Обратимся к графикам заданных нами функций принадлежности, изображенным в верхней части окна редактора функций принадлежности. Заметим, что для успешного решения поставленной задачи необходимо, чтобы ординаты максимумов этих функций совпадали с заданными значениями аргумента x . Для левой, центральной и правой функций такое условие выполнено, но две другие необходимо «подвинуть» вдоль оси абсцисс. «Передвижка» делается весьма просто: подводим курсор к нужной кривой и щелкаем левой кнопкой мыши. Кривая выбирается, окрашиваясь в красный цвет, после чего с помощью курсора ее и можно подвинуть в нужную сторону (более точную установку можно провести, изменяя числовые значения в поле Params (Параметры) – в данном случае каждой функции принадлежности соответствуют два параметра, при этом первый определяет размах кривой, а второй – положение ее центра). Для выбранной кривой, кроме этого, в поле Name можно изменять имя (завершая ввод каждого имени нажатием клавиши Enter).

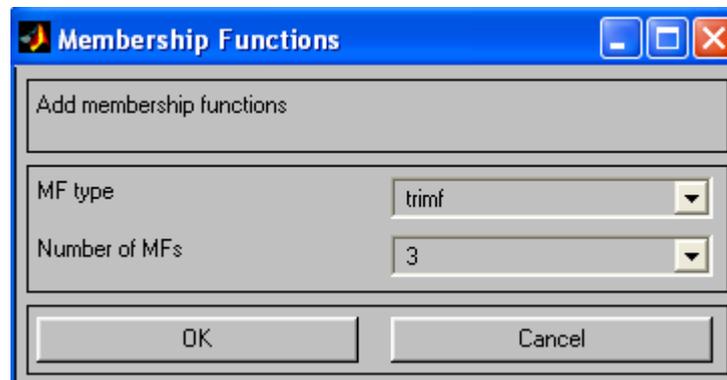


Рис. 3. Диалоговое окно задания типа и количества функций принадлежности

Выполним требуемые перемещения зависимостей и зададим всем пяти кривым новые имена, например:

- самой левой – bn,
- следующей – n,

- центральной – z ,
- следующей за ней справа – p ,
- самой левой – bp .

Нажмем кнопку Close и выйдем из редактора функций принадлежности, возвратившись при этом в окно редактора нечеткой системы (FIS Editor).

6. Сделаем однократный щелчок левой кнопкой мыши по голубому квадрату (блоку), озаглавленному output1 (выход 1). В окошке Name заменим имя output1 на y (как в пункте 2).

7. Дважды щелкнем по отмеченному блоку и перейдем к программе – редактору функций принадлежности. В позиции меню Edit выберем опцию Add MFs. Появляющееся диалоговое окно вида рис. 3 позволяет задать теперь в качестве функций принадлежности только линейные (linear) или постоянные (constant) – в зависимости от того, какой алгоритм Sugeno (1-го или 0-го порядка) мы выбираем. Если в вашем компьютере установлена версия, в которой нет данных функций принадлежности, то можно оставить по умолчанию – trimf. Это, конечно, повлияет на результат, поэтому можно поэкспериментировать, изменяя тип функций принадлежности.

В рассматриваемой задаче необходимо выбрать постоянные функции принадлежности с общим числом 4 (по числу различных значений y в табл. 1). Подтвердим введенные данные нажатием кнопки ОК, после чего произойдет возврат в окно редактора функций принадлежности.

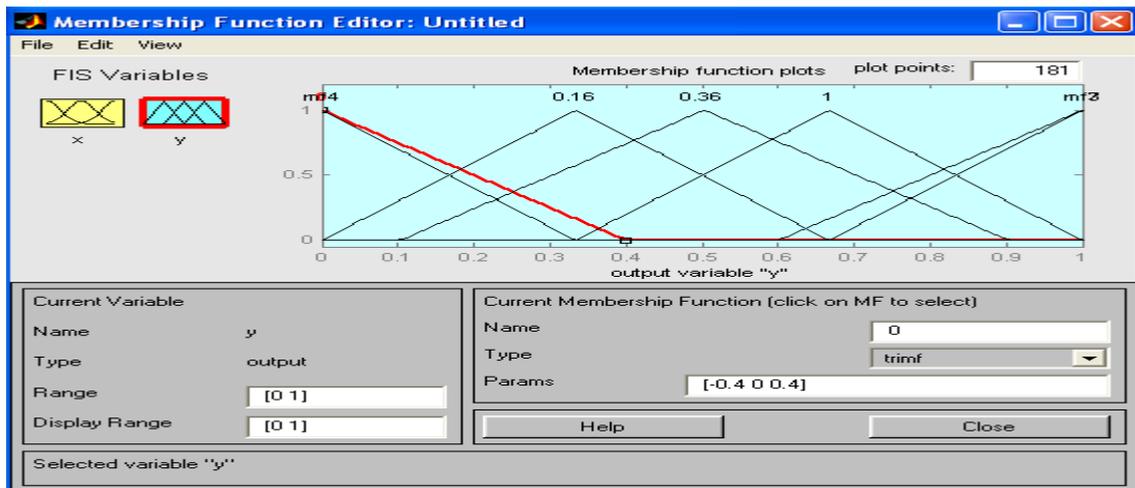


Рис. 4. Параметры функций принадлежности переменной y .

8. Диапазон (Range) изменения, устанавливаемый по умолчанию – $[0, 1]$, менять не нужно. Изменим лишь имена функций принадлежности (их графики при использовании алгоритма Sugeno для выходных переменных не приводятся), например, задав их как соответствующие числовые значения y , т.е. $0, 0.16, 0.36, 1$; одновременно эти же числовые значение введем в поле Params (рис. 4). Затем закроем окно нажатием кнопки Close и вернемся в окно FIS-редактора.

9. Дважды щелкнем левой кнопкой мыши по среднему (белому) блоку, при этом раскроется окно еще одной программы – редактора правил (Rule Editor). Введем соответствующие правила. При вводе каждого правила необходимо обозначить соответствие между каждой функцией принадлежности аргумента x и числовым значением y . Кривая, обозначенная нами bn , соответствует $x = -1$, т.е. $y = 1$. Выберем, поэтому в левом поле (с заголовком x is bn), а в правом 1 и нажмем кнопку Add rule (Добавить правило). Введенное правило появится в окне правил и будет представлять собой запись:

1. If (x is bn) then (y is 1)

Аналогично поступим для всех других значений x , в результате чего сформируется набор из 5 правил (рис. 5). Закроем окно редактора правил и возвратимся в окно FIS-редактора. Построение системы закончено и можно начать эксперименты по ее исследованию. Заметим, что большинство опций выбиралось нами по умолчанию.

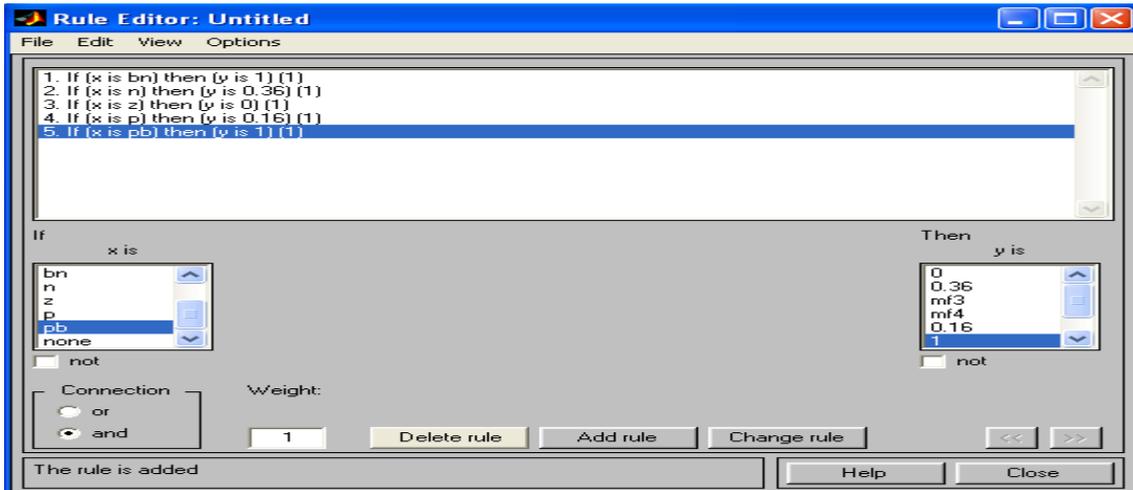


Рис. 5. Окно редактора правил.

10. Предварительно сохраним на диске (используя пункты меню File/Save to disk as...) созданную систему под каким-либо именем, например, Proba.

11. Выберем позицию меню View. Как видно из выпадающего при этом подменю, с помощью пунктов Edit membership functions и Edit rules можно совершить переход к двум выше рассмотренным программам – редакторам функций принадлежности и правил (то же можно сделать и нажатием клавиш Ctrl+2 или Ctrl+3). Но сейчас нас будут интересовать два других пункта – View rules (Просмотр правил) и View surface (Просмотр поверхности). Выберем пункт View rules, при этом откроется окно (рис. 6) еще одной программы – просмотра правил (Rule Viewer).

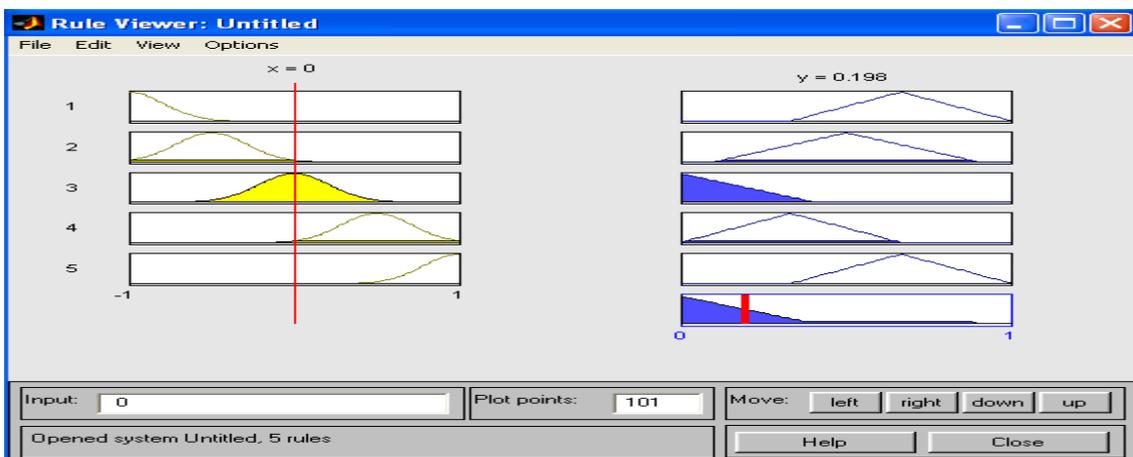


Рис. 6. Окно просмотра правил.

12. В правой части окна в графической форме представлены функции принадлежности аргумента x , в левой – переменной выхода y с пояснением механизма принятия решения. Красная вертикальная черта, пересекающая графики в правой части окна, которую можно перемещать с помощью курсора, позволяет изменять значения переменной входа (это же можно делать, задавая числовые значения в поле Input (Вход)), при этом соответственно изменяются значения y в правой верхней части окна. Зададим, например, $x = 0.5$ в поле Input и нажмем затем клавишу ввода (Enter). Значение y сразу изменится и станет равным 0.202. Таким образом, с помощью построенной модели и окна просмотра правил можно решать задачу интерполяции, т.е. задачу, решение которой и требовалось найти. Изменение аргумента путем перемещения красной вертикальной линии очень наглядно демонстрирует, как система определяет значения выхода.

13. Закроем окно просмотра правил и выбором пункта меню View/View surface перейдем к окну просмотра поверхности отклика (выхода), в нашем случае – к просмотру кривой $y(x)$ (рис. 7). Смоделированное по таблице данных (табл. 1) отображение не совсем адекватно функции x^2 , потому что число экспериментальных точек небольшое, и параметры функций принадлежности (для x) выбраны, неоптимальным образом.

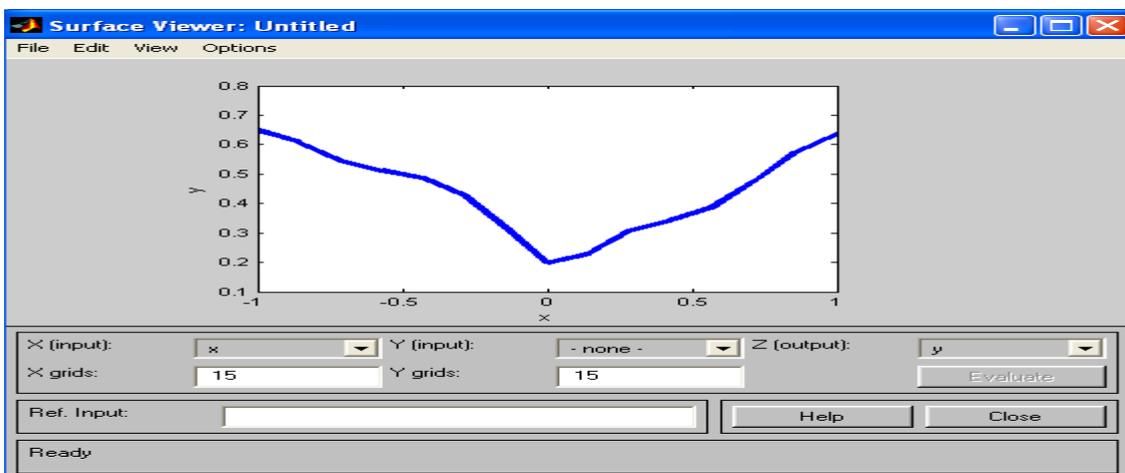


Рис. 7. Окно просмотра поверхности отклика.

С помощью вышеуказанных программ-редакторов на любом этапе проектирования нечеткой модели в нее можно внести необходимые коррективы, вплоть до задания какой-либо особенной пользовательской функции принадлежности. Из опций, устанавливаемых в FIS-редакторе по умолчанию можно отметить:

- логический вывод организуется с помощью операции умножения (prod);
- композиция – с помощью операции логической суммы (вероятностного ИЛИ, probor);
- приведение к четкости – дискретным вариантом центроидного метода (взвешенным средним, wtaver). Используя соответствующие поля в левой нижней части окна FIS-редактора, данные опции можно, при желании, изменить.

Задание на лабораторную работу

Разработать нечеткую систему, отображающую зависимость между переменными x и y , заданную с помощью табл. 2. По результатам работы определите тип кривой.

Таблица 2

Вариант							
1	x	-1	-0.5	0	0.2	1	
	y	1	0.25	0	0.4	1	
2	x	-1	-0.6	0.2	0.4	1	
	y	-1	-	5	2.5	1	
			1.67				
3	x	-1	-0.5	0	0.3	1	
	y	-1	-	0	0.27	1	
			0.13				

4	x	-1	-	0	0.3	1	
			0.6				
	y	0	0.8	1	0.95	0	
5	x	-2	-	0	0.56	2	
			0.35				
	y	1	0.56	0.1	0.3	1	
6	x	-1	-0.5	0.2	0.3	1	
	y	-1	-	5	2.12	1	
			1.34				
7	x	-2	-0.3	0	0.4	2	
	y	-1	-	-	0.24	1	
			0.12	0.23			
8	x	-1	-	0	0.3	1	
			0.6				
	y	0	0.8	1	0.95	0	
9	x	-1	-0.6	0	0.3	1	
	y	1	0.36	0	0.9	1	

3. Контрольные вопросы

1. Дайте определение лингвистической переменной
2. Чем четкая переменная отличается от нечеткой?
3. В чем заключается процедура фаззификации и дефаззификации?
4. Чем модель Мамдани отличается от модели Сугено?
5. Чем функция принадлежности отличается от характеристической функции?

4. Содержание отчета

1. Цель работы;
2. Задание;
3. Краткое описание действий;
4. Окна и графики по всем пунктам программы;
5. Выводы по работе.
6. Ответы на контрольные вопросы.

Лабораторная работа 6

Нейросетевая аппроксимация модели

- 1. Цель работы.** Изучить процедуры идентификации моделей и понять необходимость аппроксимации функций входов и выходов при идентификации математических моделей. Для этой цели целесообразно применить новый подход в виде нейронных сетей. Научиться работать с нейронной сетью прямой передачи данных и с алгоритмом обучения методом обратного распространения ошибки для аппроксимации нелинейной функции.

2. Краткие теоретические сведения

При построении математических моделей большого числа объектов достаточно эффективным оказывается использование методов идентификации - процедур построения адекватной в регламентированном смысле модели какого-либо объекта по экспериментальным данным (реализациям входных и выходных процессов).

Идентифицированные модели используются в дальнейшем для анализа и оптимизации процессов функционирования, диагностики, управления различных систем, создания математического обеспечения соответствующих САПР, АСНИ. В связи с этим современному специалисту необходимы знания и навыки построения достаточно простых и адекватных математических

моделей систем, основываясь на реальных данных. Настоящие лабораторные работы ориентированы на ознакомление студентов с основами теории идентификации систем, развитие навыков построения моделей сложных систем. Под идентификацией объекта понимают построение его математической модели по данным о реакции объекта на известные внешние воздействия. Задача идентификации формулируется следующим образом (рис.1).

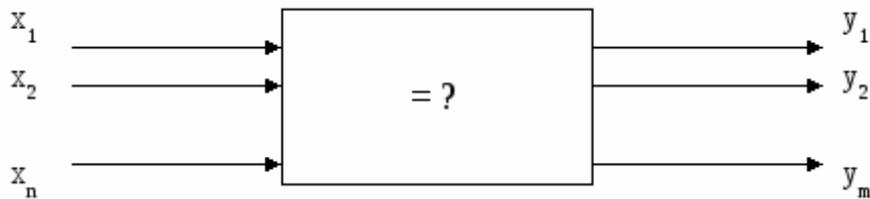


Рис. 1. Принципиальная схема задачи идентификации

Пусть в результате каких-либо экспериментов над некоторым объектом замерены его входные переменные $X = (x_1, x_2, \dots, x_n)$ и выходные переменные $Y = (y_1, y_2, \dots, y_m)$. Требуется определить вид (структуру) и параметры некоторого оператора \hat{A} , ставящего в соответствие переменные X и Y : $Y = \hat{A}(X)$. Различают задачи идентификации в широком (структурная идентификация) и узком (параметрическая идентификация) смысле. В первом случае считаются неизвестными структура и параметры оператора \hat{A} , во втором - лишь параметры этого оператора.

В качестве «объекта идентификации» в общем случае могут рассматриваться механическая, электрическая, экономическая или биологическая система, замкнутые или разомкнутые (с обратной связью) системы управления и т. д. Методы идентификации таких разнообразных по

назначению, виду и структуре объектов естественно различаются, хотя существует и ряд общих закономерностей. В практике идентификации обычно используют совокупность различных методов, применяя их в порядке возрастания адекватности и точности получаемых моделей.

Будем считать, что в общем случае в результате экспериментов получена информация о состояниях выходных переменных Y_j и входных переменных X_i , от которых предположительно зависят Y_j . Результатом наблюдений является таблица экспериментальных данных:

$$T_{X,Y} = \begin{pmatrix} X_{11} & X_{21} & \dots & X_{n1} & Y_{11} & Y_{21} & \dots & Y_{m1} \\ X_{12} & X_{22} & \dots & X_{n2} & Y_{12} & Y_{22} & \dots & Y_{m2} \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ X_{1N} & X_{2N} & \dots & X_{nN} & Y_{1N} & Y_{2N} & \dots & Y_{mN} \end{pmatrix}$$

где N – количество опытов (повторений эксперимента), n – число входных переменных, m – число выходных переменных, X_{ik} – значение i -ой входной переменной для k -го опыта, Y_{jk} – значение j -той выходной переменной для k -го опыта.

Требуется для каждой выходной переменной Y_j построить аналитическую зависимость $y_j = \hat{A}_j(X)$, адекватно описывающую имеющиеся экспериментальные данные (в дальнейшем индекс при y будем опускать, предполагая, что выходные переменные друг от друга не зависят, и без ограничения общности достаточно рассмотреть случай одномерного вектора Y : $y = \hat{A}(X)$).

Интерполяционные методы обработки данных в точности воспроизводят значения данной функции в узлах интерполяции. Однако в ряде случаев выполнение этого условия затруднительно или даже нецелесообразно:

1. Если заданные величины x и y являются экспериментальными данными, то могут содержать в себе существенные ошибки, так как получены в результате измерений или наблюдений. Поэтому построение аппроксимирующего

многочлена, воспроизводящего в точности заданное значение функции, означало бы тщательное копирование допущенных при измерениях ошибок.

2. Если имеются точные значения функции в некоторых точках, но число таких точек n весьма велико, то интерполяционный многочлен будет очень высокой степени (при условии, что разности не будут постоянными).

Поэтому возникает задача аппроксимации – построения многочлена некоторой вполне определенной степени, но меньшей чем $n - 1$, который хотя и не дает точных значений функции в узлах интерполяции, но достаточно близко к ним подходит.

В лабораторной работе необходимо получить значения экспериментальных данных из таблицы и аппроксимировать эти зависимости нейронной сетью в программной среде МатЛаб..

Для реализации процедуры аппроксимации использовать нейронную сеть с прямой передачей сигнала (с прямой связью), то есть сеть, в которой сигналы передаются только в направлении от входного слоя к выходному, и элементы одного слоя связаны со всеми элементами следующего слоя. Важнейшим для реализации нейронных сетей является определение алгоритма обучения сети.

В настоящее время одним из самых эффективных и обоснованных методов облучения нейронных сетей является *алгоритм обратного распространения ошибки*, который применим к *однонаправленным многослойным сетям*. В многослойных нейронных сетях имеется множество скрытых нейронов, входы и выходы которых не являются входами и выходами нейронной сети, а соединяют нейроны внутри сети, то есть *скрытые нейроны*. Пронумеруем выходы нейронной сети индексом $j = 1, 2, \dots, n$, а обучающие примеры индексом $M = 1, 2, \dots, M_0$. Тогда в качестве целевой функции можно выбрать функцию ошибки как сумму квадратов расстояний между реальными выходными состояниями y_{jM} нейронной сети, выдаваемых сетью на входных данных примеров, и правильными значениями функции d_{jM} , соответствующими этим примерам. Пусть $\mathbf{x} = \{x_i\}$ – столбец входных

значений, где $i=1,2,\dots,n$. Тогда $\mathbf{y} = \{y_j\}$ – выходные значения, где $j=1,2,\dots,m$. В общем случае $n \neq m$. Рассмотрим разность $y_{jM} - d_{jM}$, где d_{ji} – точное (правильное) значение из примера. Эта разность должна быть минимальна. Введем расстояния согласно евклидовой метрике, определив норму

$$\|\mathbf{y} - \mathbf{d}\| = \sqrt{(\mathbf{y} - \mathbf{d}, \mathbf{y} - \mathbf{d})}.$$

Пусть целевая функция имеет вид

$$E = \frac{1}{2} \sum_{j,M} (y_{j,M} - d_{j,M})^2.$$

Коэффициент $\frac{1}{2}$ выбран из соображений более короткой записи последующих формул. Задача обучения нейронной сети состоит в том, чтобы найти такие коэффициенты $w_{\beta k}$, при которых достигается минимум $E(\mathbf{w})$ ($E \geq 0$).

На рис. 2 показана архитектура нейронной сети с прямой передачей сигнала.

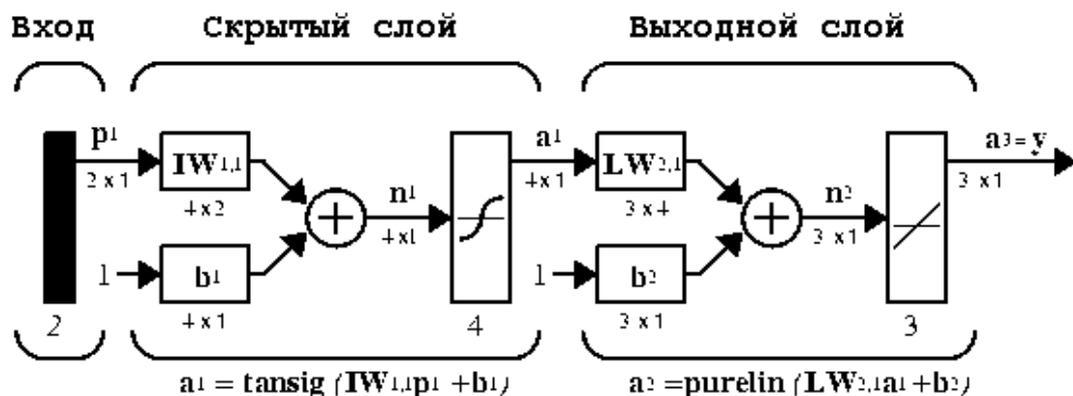


Рис. 2. Схема архитектуры нейронной сети с прямой передачей сигнала

Здесь приняты обозначения, используемые в [1], а именно, \mathbf{p}^1 – вектор входа, $IW^{i,j}$, $LW^{i,j}$ – матрицы весов входа и выхода, \mathbf{b}^i – смещение, \mathbf{a}^i – выход слоя, \mathbf{y} – выход сети, $tansig$ (гиперболическая тангенциальная), $purelin$ (линейная) – соответствующие функции активации.

Веса и смещения определяются с помощью алгоритма обратного распространения ошибок [3].

Обучение сети обратного распространения требует выполнения следующих операций:

1. Выбрать очередную обучающую пару из обучающего множества; подать входной вектор на вход сети.
2. Вычислить выход сети.
3. Вычислить разность между выходом сети и требуемым выходом (целевым вектором обучающей пары).
4. Скорректировать веса сети так, чтобы минимизировать ошибку.
5. Повторять шаги с 1 по 4 для каждого вектора обучающего множества до тех пор, пока ошибка на всем множестве не достигнет приемлемого уровня.

3. Пример решения задачи

Выполнение лабораторной работы состоит из следующих этапов: прежде всего, необходимо оцифровать график функции $y=f(x)$, то есть получить ряд соответствующих значений по горизонтальной и вертикальной осям.

В примере, показанном на рис. 2 были получены два массива, каждый из которых состоит из 15 значений. По горизонтальной оси – **[0.10 0.31 0.51 0.72 0.93 1.14 1.34 1.55 1.76 1.96 2.17 2.38 2.59 2.79 3.00]**. По вертикальной оси – **[0.1010 0.3365 0.6551 1.1159 1.7632 2.5847 3.4686 4.2115 4.6152 4.6095 4.2887 3.8349 3.4160 3.1388 3.0603]**.

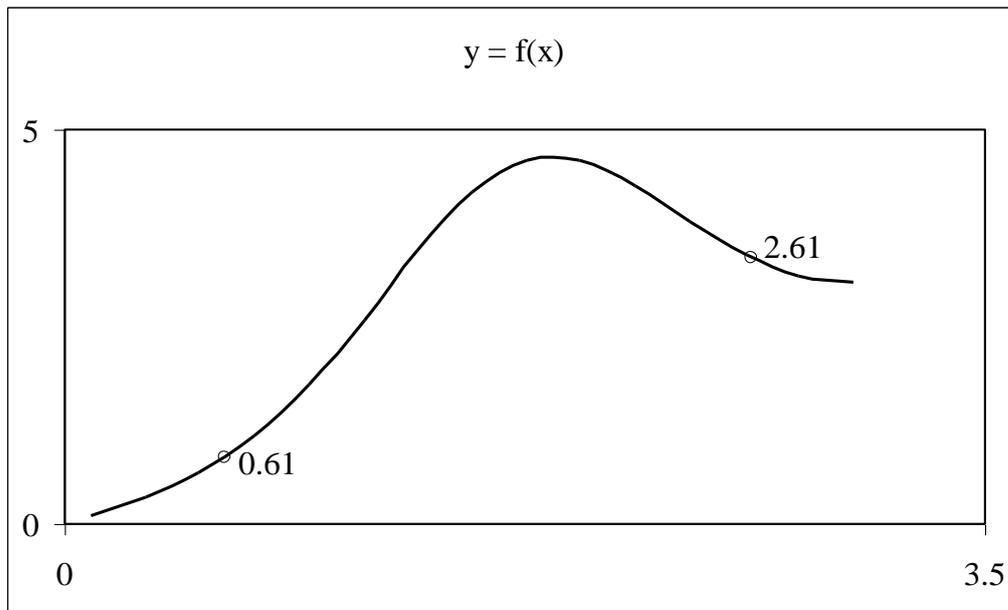


Рис. 2. Пример зависимости для функции одной переменной

Программа создания, обучения нейронной сети и вывода результатов.

```

x=[0.10 0.31 0.51 0.72 0.93 1.14 ...
1.34 1.55 1.76 1.96 2.17 2.38 ...
2.59 2.79 3.00];
y=[0.1010 0.3365 0.6551 1.1159 1.7632 2.5847 ...
3.4686 4.2115 4.6152 4.6095 4.2887 3.8349 ...
3.4160 3.1388 3.0603];
net=newff([0 3],[5,1],{'tansig','purelin'},'trainbfg');
net.trainParam.epochs=300;
net.trainParam.show=50;
net.trainParam.goal=1.37e-2;
[net,tr]=train(net,x,y);
an=sim(net,x);
plot(x,y,'+r',x,an,'-g'); hold on;
xx=[0.61 2.61];
v=sim(net,xx)
plot(xx,v,'ob','MarkerSize',5,'LineWidth',2)

```

В результате выполнения программы получаются следующие результаты, отражённые на рис. 3 и 4:

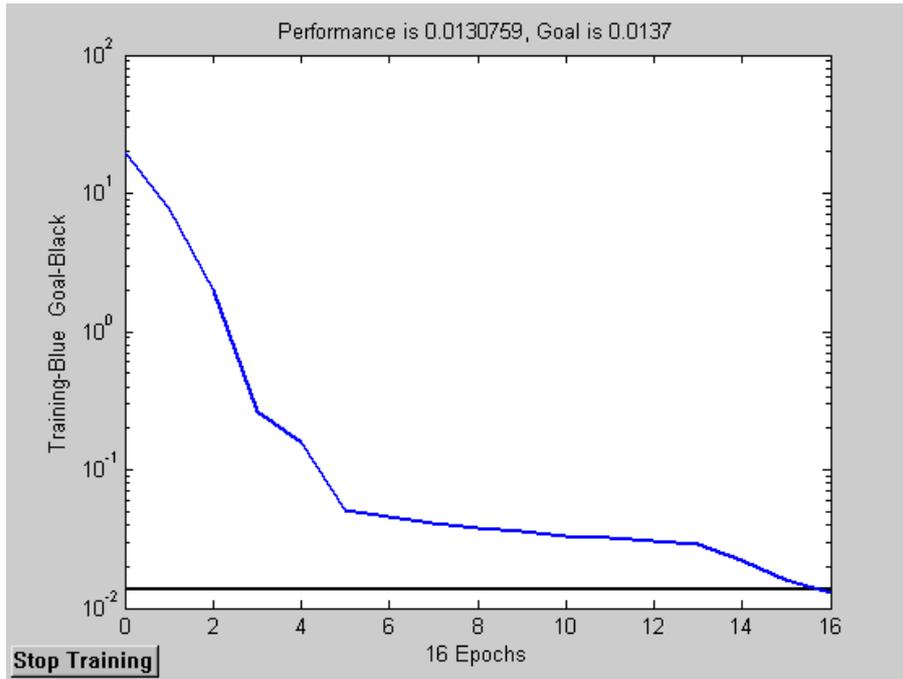


Рис. 3. Характеристика точности обучения в зависимости от числа эпох обучения

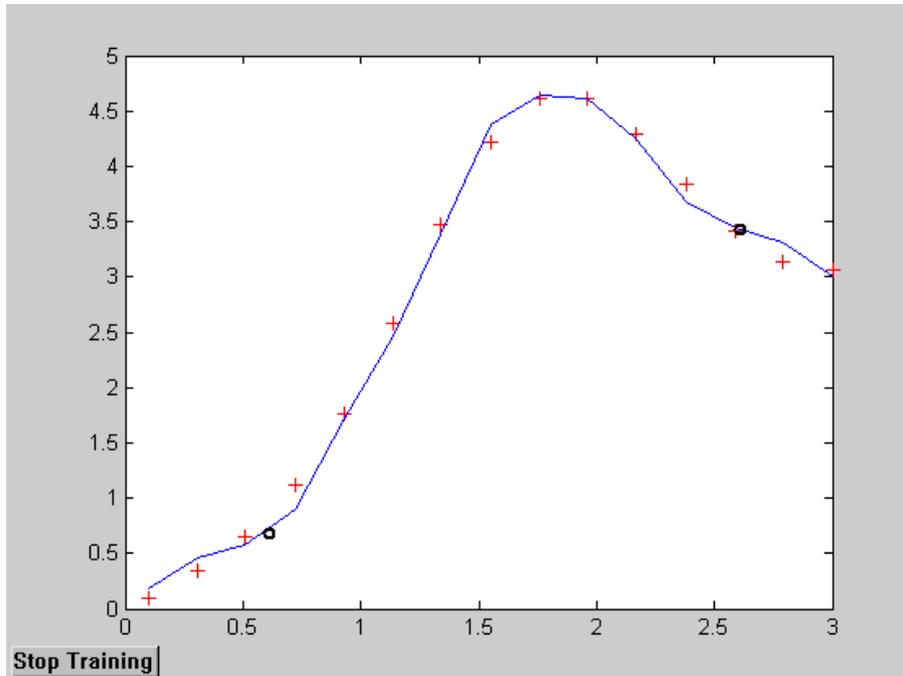


Рис. 4. Результаты моделирования сети: + - исходные данные; сплошная линия и символ «o» – результаты моделирования всей зависимости и в контрольных точках

В массиве \mathbf{v} содержатся приближённые значения для двух контрольных точек, указанных на графике (рис. 2) $\mathbf{xx}=[0.61 \ 2.61]$. При данных параметрах сети получены значения: $\mathbf{v} = [1.05 \ 3.35]$. Сравнив эти приближённые значения с точными значениями $[0.85 \ 3.37]$, можно сделать вывод о корректности построения нейронной сети.

4. Содержание отчета

1. Исходные данные своего варианта (рис. 1);
2. Текст программы с подробными комментариями;
3. Характеристику точности обучения нейронной сети
4. Результаты моделирования;
5. Сопоставление результатов в контрольных точках;
6. Краткие выводы о результатах работы.
7. Ответы на вопросы.

5. Варианты заданий

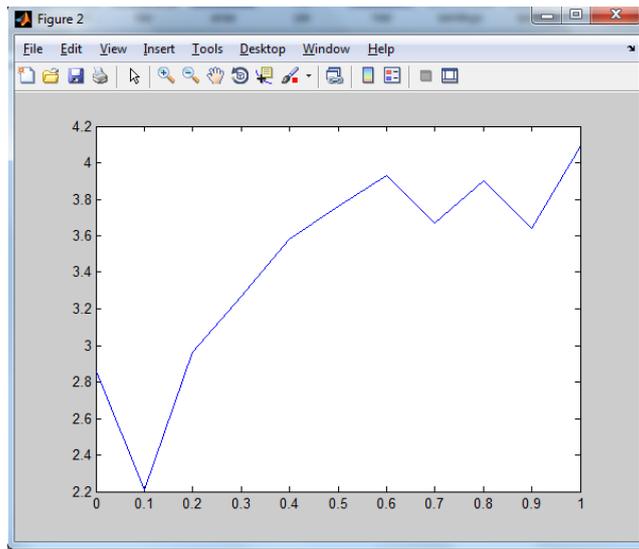
Выбор варианта. По последней цифре в зачетной книжке, если больше 30, то сумма этих цифр.

Создайте таблицу 1 экспериментальных данных:

$x_i = a + h \cdot i, i = 0, 1, \dots, 10, h = (b - a) / 10$ К на отрезке $[a, b]$.

Вариант

	i	y	$[a, b]$
1	2.86; 2.21; 2.96; 3.27; 3.58; 3.76; 3.93; 3.67; 3.90; 3.64; 4.09		[0, 1]
2	1.14; 1.02; 1.64; 1.64; 1.96; 2.17; 2.64; 3.25; 3.47; 3.89; 3.36		[-1, 1]
3	4.70; 4.64; 4.57; 4.45; 4.40; 4.34; 4.27; 4.37; 4.42; 4.50; 4.62		[2, 4]
4	0.43; 0.99; 2.07; 2.54; 1.67; 1.29; 1.24; 0.66; 0.43; 0.35; 0.70		[2, 4]
5	1.55; 1.97; 1.29; 0.94; 0.88; 0.09; 0.02; 0.84; 0.81; 0.09; 0.15		[1, 4]
6	3.24; 1.72; 1.95; 2.77; 2.47; 0.97; 1.75; 1.55; 0.12; 0.70; 1.19		[0, 4]
7	2.56; 1.92; 2.85; 2.94; 2.39; 2.16; 2.51; 2.10; 1.77; 2.28; 1.70		[-1, 2]
8	1.77; 0.92; 2.21; 1.50; 3.21; 3.46; 3.70; 4.02; 4.36; 4.82; 4.03		[-1, 3]



a)

Рис. 5. График исходной функции

Создание и обучение нейронной сети:

```

x=[00.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1];
y=[2.86 2.21 2.96 3.27 3.58 3.76 3.93 3.67 3.90 3.64 4.09];
net=newff([0 3],[10,1],{'tansig','purelin'},'trainbfg');
net.trainParam.epochs=300;
net.trainParam.show=50;
net.trainParam.goal=1.37e-2;
[net,tr]=train(net,x,y);
an=sim(net,x);
plot(x,y,'+r',x,an,'-g'); hold on;
xx=[0.1850.86];
v=sim(net,xx)
plot(xx,v,'ob','MarkerSize',5,'LineWidth',2)

```

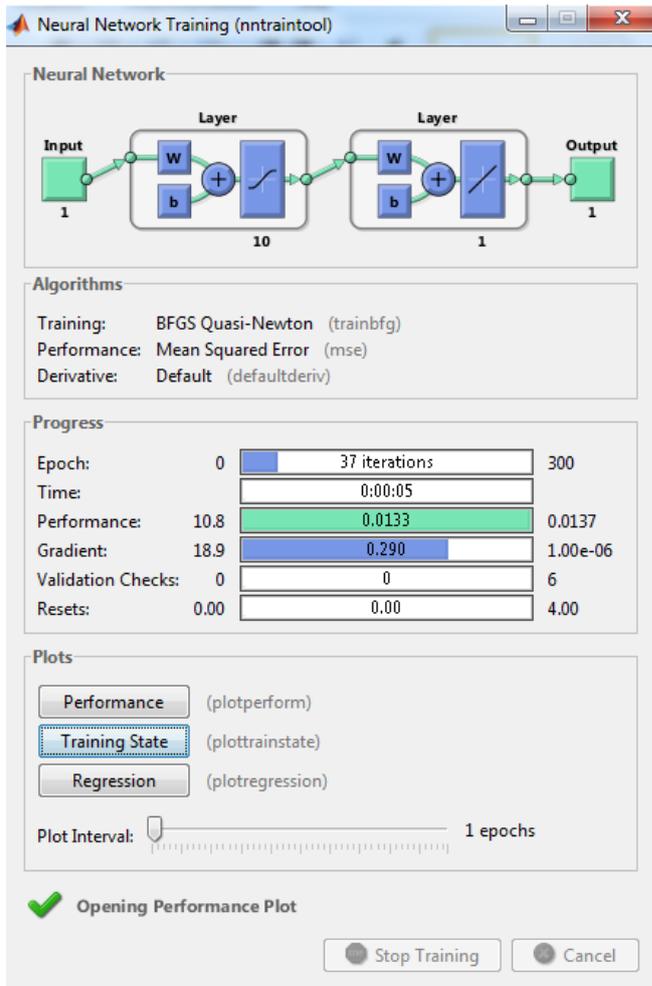


Рис.6. Обучение сети

В процессе обучения сети получился график зависимости характеристики точности обучения сети от количества эпох (циклов), и вычисление среднеквадратичной ошибки сети составляет 0,013305 за 37 циклов:

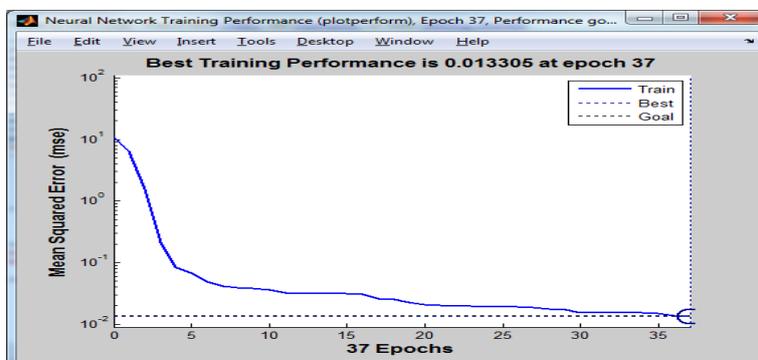


Рис.7. Характеристика точности обучения в зависимости от количества эпох обучения

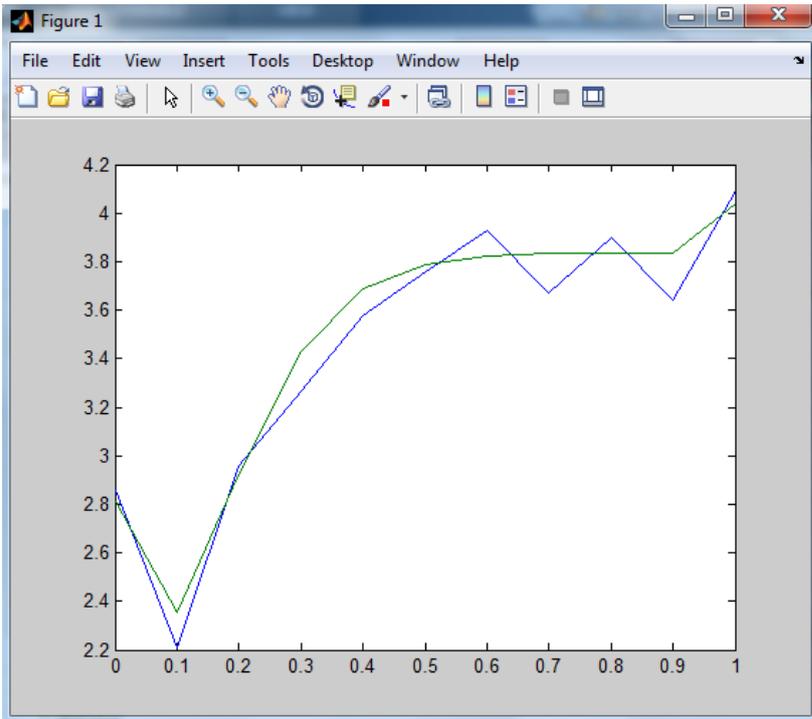


Рис. 8. Сравнение графиков исходной функции и аппроксимации

Аппроксимируем входящий набор точек методом МНК:

Результат для набора а) представлен ниже:

Аналогичные действия проделываем для другого набора точек:

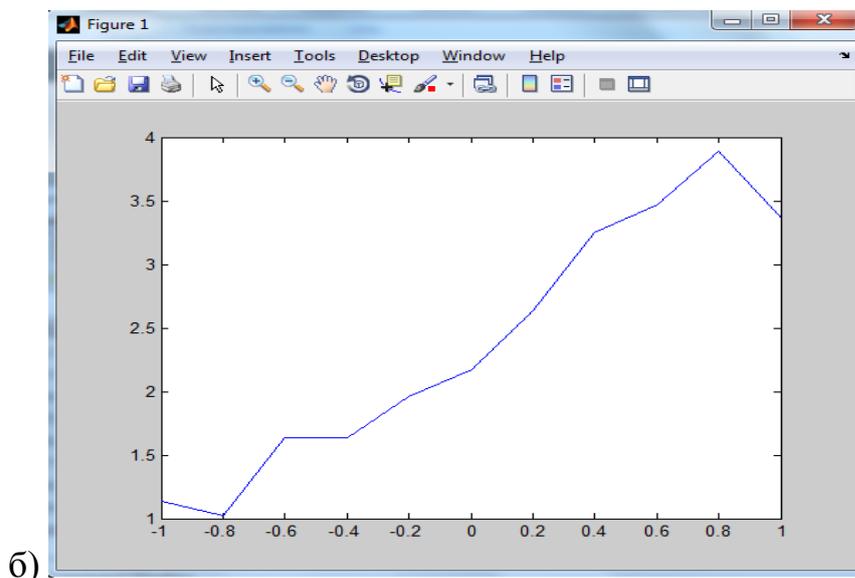


Рис. 9. График исходной функции

Вывод: установлено по полученным результатам аппроксимации заданного набора значений функции, нейронная сеть намного лучше аппроксимирует исходные значения функции, чем МНК.

Список рекомендованной литературы

1. Замятин Н.В. Рынки ИКТ и организация продаж : учеб. пособие. –Томск, 2015. – 202 с.
2. Соловьев В. И. Стратегия и тактика конкуренции на рынке программного обеспечения : Опыт экономико_математического моделирования: монография / В. И. Соловьев. — М.: Вега_Инфо, 2010. — 200 с.
3. Замятин Н.В. Методические указания к выполнению лабораторных и практических работ по дисциплине «Рынки ИКТ и организация продаж». - Томск : ТУСУР, каф. АОИ, 2015 - 100 с. [Электронный ресурс]: сайт кафедры АОИ. – ГКД
4. AnyLogic User's Manual. XJ Technologies : [электрон. ресурс]. Режим доступа : <http://www.xjtek.com>
5. AnyLogic Tutorial. XJ Technologies : [электрон. ресурс]. Режим доступа : <http://www.xjtek.com>.
6. Дьяконов В.П. MATLAB 6/6.1/6.5 + SIMULINK 4/5 в математике и моделировании.– М.: СОЛОН-Пресс, 2003 – 565 с. 7. Круглов В. В., Длин М.И., Голунов Р. Ю. Нечёткая логика и искусственные нейронные сети: Учеб. Пособие. – М.: Издательство Физико-математической литературы, 2001. – 224 с.
38. Замятин Н.В. Нечеткая логика и нейронные сети. Учебное пособие. Изд-во Эль-контент.- Томск. 2015-205 с.

