

Международный консорциум «Электронный университет»

*Московский государственный университет экономики,
статистики и информатики*

Евразийский открытый институт

Н.Н. Снетков

**Имитационное моделирование
экономических процессов**

Учебно-практическое пособие

Москва 2008

УДК 519.86
ББК 65.050
С 534

Снетков Н.Н. **Имитационное моделирование экономических процессов:** Учебно-практическое пособие. – М.: Изд. центр ЕАОИ, 2008. – 228 с.

ISBN 978-5-374-00079-5 © Снетков Н.Н., 2008
© Евразийский открытый институт, 20087

Содержание

Цель и задачи курса	8
Введение	11
Раздел I. Теоретические основы имитационного моделирования.....	13
<i>Глава 1. Основные понятия теории моделирования экономических систем и процессов</i>	<i>13</i>
§1. Основы принятия решений относительно создания, совершенствования, развития экономических систем....	13
§2. Основы имитационного моделирования	20
2.1. Понятие модели.....	20
2.2. Классификация моделей	21
2.3. Последовательность разработки математических моделей.....	24
2.3.1. Определение цели моделирования.....	25
2.3.2. Построение концептуальной модели.....	26
2.3.3. Разработка алгоритма модели системы	29
2.3.4. Разработка программы модели системы	29
2.3.5. Планирование модельных экспериментов и проведение машинных экспериментов с моделью системы.....	30
<i>Глава 2. Математические схемы моделирования экономических систем</i>	<i>31</i>
§1. Классификация моделируемых систем	31
§2. Математические схемы (модели).....	34
<i>Глава 3. Моделирование случайных событий и величин</i>	<i>38</i>
§1. Моделирование случайных событий	41
1.1. Моделирование простого события	41
1.2. Моделирование полной группы несовместных событий	47
§2. Моделирование случайных величин	49
2.1. Моделирование дискретной случайной величины..	49
2.2. Моделирование непрерывных случайных величин.....	50

2.2.1. Метод обратной функции.....	50
2.2.2. Моделирование случайных величин с показательным распределением	50
2.2.3. Моделирование случайных величин с равномерным распределением.....	51
2.2.4. Моделирование случайных величин с нормальным распределением.....	52
2.2.5. Моделирование случайных величин с усеченным нормальным распределением ...	54
2.2.6. Моделирование случайных величин с произвольным распределением.....	56
2.2.7. Моделирование случайных величин с заданными параметрами средствами Matlab ...	58
Раздел II. Концепция и возможности объектно- ориентированной моделирующей системы.....	63
<i>Глава 4. Общие сведения о MATLAB/SIMULINK.</i>	
<i>Библиотека блоков SIMULINK.....</i>	<i>63</i>
§1. Запуск MATLAB, интерфейс	64
§2. Editor/ debugger – редактор\отладчик программ	67
§3. Простые вычисления в командном режиме	69
§4. Введение в Simulink.....	70
§5. Работа с Simulink.....	71
§6. Обзорщик разделов библиотеки Simulink.....	73
§7. Создание модели	75
§8. Окно модели	78
§9. Основные приемы подготовки и редактирования модели	81
§10. Библиотека блоков SIMULINK.....	87
10.1. Sources – источники сигналов	87
10.2. Sinks – приемники сигналов	88
10.2.1. Осциллограф Scope	88
10.2.2. Цифровой дисплей Display	93
10.3. Continuous – аналоговые блоки	95
10.3.1. Интегрирующий блок Integrator	95

10.3.2. Блок фиксированной задержки сигнала Transport Delay.....	98
10.3.3. Блок управляемой задержки сигнала Variable Transport Delay.....	99
10.4. Nonlinear – нелинейные блоки	100
10.4.1. Блок ограничения Saturation	100
10.4.2. Блок переключателя Switch.....	102
10.4.3. Блок ручного переключателя Manual Switch	103
10.5. Math – блоки математических операций.....	103
10.5.1. Блок вычисления суммы Sum	103
10.5.2. Усилители Gain и Matrix Gain.....	105
10.5.3. Блок вычисления операции отношения Relational Operator.....	107
10.6. Signal&Systems – блоки преобразования сигналов и вспомогательные блоки	109
10.6.1. Мультиплексор (смеситель) Mux.....	109
10.6.2. Демультиплексор (разделитель) Demux	110
10.7. Function & Tables – блоки функций и таблиц.....	112
10.7.1. Блок задания функции Fcn	112
10.7.2. Блок задания функции MATLAB Fcn	114
10.8. Этапы моделирования	115
Глава 5. Управление модельным временем	117
§1. Виды представления времени в модели	117
§2. Изменение времени с постоянным шагом.....	118
§3. Продвижение времени по особым состояниям	121
§4. Моделирование параллельных процессов.....	122
§5. Управление модельным временем в matlab.....	128
§6. Установка параметров вывода выходных сигналов моделируемой системы output options (параметры вывода)	141
§7. Установка параметров обмена с рабочей областью....	142
§8. Установка параметров диагностирования модели..	143

Раздел III. Основные правила моделирования	145
<i>Глава 6. Классификация математических моделей экономических систем</i>	<i>145</i>
§1. Общие экономические модели	145
§2. Модели управления предприятием	149
<i>Глава 7. Моделирование процессов обслуживания заявок в условиях отказов.....</i>	<i>153</i>
<i>Глава 8. Планирование модельных экспериментов</i>	<i>160</i>
§1. Цели планирования экспериментов.....	160
§2. Стратегическое планирование имитационного эксперимента.....	162
§3. Тактическое планирование эксперимента.....	166
§4. Возможности Matlab/Simulink по планированию и реализации модельных экспериментов	169
4.1. Разработка планов экспериментов.....	169
4.2. Проведение имитационных экспериментов с использованием файлов сценариев	172
<i>Глава 9. Примеры построения имитационных моделей....</i>	<i>174</i>
§1. Имитационная модель циклов роста и падений в экономике (кризисов).....	174
1.1. Постановка задачи на моделирование.....	174
1.2. Построение концептуальной модели	174
1.3. Математическая модель.....	175
§2. Использование имитационного моделирования для поиска оптимальной ставки налогообложения на прибыль.....	178
2.1. Постановка задачи на моделирование.....	178
2.2. Построение концептуальной модели	179
2.3. Математическая модель.....	180
2.4. Компьютерная модель в программе Simulnk	181
2.5. Исходные данные для параметров, переменных и показателей модели	183
2.6. Математическая схема модели и метод решения ...	183
2.7. Средства управления экспериментом.....	183

2.8. Программа управления имитационным экспериментом	184
§3. «Паутинообразная» модель фирмы (равновесие на конкурентном рынке)	185
3.1. Постановка задачи на моделирование.....	185
3.2. Построение модели.....	188
Практикум.....	190
Практическое занятие 1.....	190
Практическое занятие 2.....	196
Практическое занятие 3.....	201
Практическое занятие 4.....	206
Практическое занятие 5.....	207
Практическое занятие 6.....	209
Контрольные работы (для заочного отделения)	211
Темы лабораторных (семестровых) работ.....	211
Итоговые вопросы	212
Глоссарий.....	214
Список рекомендуемой литературы.....	227

Цель и задачи курса

Целью изучения дисциплины является формирование у студентов теоретических знаний и практических навыков по применению методов имитационного моделирования в экономике, управлении и бизнесе. В процессе изучения курса студенты знакомятся со средствами имитационного моделирования процессов функционирования экономических систем, овладевают методами имитационного моделирования, типовыми этапами моделирования процессов, образующих «цепочку»: построение концептуальной модели и ее формализация – алгоритмизация модели и ее компьютерная реализация – имитационный эксперимент и интерпретация результатов моделирования; овладевают практическими навыками реализации моделирующих алгоритмов для исследования характеристик и поведения сложных экономических систем.

В результате освоения содержания дисциплины студент должен:

а) знать :

- основы теории и практики имитационного моделирования экономических процессов;
- основные классы моделей систем предметной области, технологию их моделирования;
- принципы построения моделей процессов функционирования экономических систем, методы формализации и алгоритмизации, возможности реализации моделей с использованием программно-технических средств современных ЭВМ.

б) уметь:

- использовать метод имитационного моделирования при исследовании, проектировании и эксплуатации экономических систем;
- разрабатывать схемы моделирующих алгоритмов процессов и систем, реализовывать модели с использованием пакета прикладных программ моделирования Matlab, Simulink.

Требования, предъявляемые к слушателю курса

Для изучения курса «Имитационное моделирование экономических процессов» студент должен знать теорию систем и системного анализа, экономику, математику, теорию вероятностей математической статистики, программирование, а также иметь навыки пользователя ПК.

Дисциплина опирается на ранее изученные дисциплины «Экономика», «Математика. Математический анализ», «Теория вероятностей», «Теория статистики», «Математика. Линейная алгебра», «Математика. Дискретная математика», «Численные методы», «Информатика и программирование», «Высокоуровневые методы информатики и программирования», «Теория экономических информационных систем», «Методы оптимизации», «Теория систем и системный анализ» и используется при изучении дисциплин: «Проектирование информационных систем», «Технология внедрения корпоративных информационных систем», «Реинжиниринг бизнес-процессов».

Виды работ, выполняемых в ходе курса:

- Участие в различных формах обучения:
 - посещение лекционных и практических занятий;
 - совместная работа;
 - получение информации от преподавателя и общение с ним для решения возникающих вопросов, связанных с содержанием дисциплины;
 - самостоятельная работа.
- Изучение теоретического материала.
- Самостоятельное изучение отдельных разделов теоретического материала с использованием электронных ресурсов.
- Выполнение заданий практической части курса.
- Ответы на вопросы для самопроверки по окончании изучения курса.
- Прохождение итогового контроля.

Для успешного прохождения курса необходимо:

- Участие в теоретической и практической части изучения курса.
- Чтение рекомендуемой литературы по соответствующим темам.
- Выполнение всех практических заданий.
- Прохождение самотестирования по темам курса в период его изучения.
- Прохождение самотестирования по всему курсу.

Для прохождения итогового контроля необходимо

Успешно ответить на вопросы итогового теста в ИОС «Прометей» и сдать экзамен.

Введение

Имитационное моделирование – наиболее мощный и универсальный метод исследования и оценки эффективности систем, поведение которых зависит от воздействия случайных факторов. К таким системам можно отнести и летательный аппарат, и популяцию животных, и предприятие, работающее в условиях слаборегулируемых рыночных отношений.

В основе имитационного моделирования лежит статистический эксперимент (метод Монте-Карло), реализация которого практически невозможна без применения средств вычислительной техники. Поэтому любая имитационная модель представляет собой в конечном счете более или менее сложный программный продукт.

Конечно, как и любая другая программа, имитационная модель может быть разработана на любом универсальном языке программирования, даже на языке Ассемблера. Однако на пути разработчика в этом случае возникают следующие проблемы:

- требуется знание не только той предметной области, к которой относится исследуемая система, но и языка программирования, причем на достаточно высоком уровне;
- на разработку специфических процедур обеспечения статистического эксперимента (генерация случайных воздействий, планирование эксперимента, обработка результатов) может уйти времени и сил не меньше, чем на разработку собственно модели системы.

И наконец, еще одна, пожалуй, важнейшая проблема. Во многих практических задачах интерес представляет не только (и не столько) количественная оценка эффективности системы, сколько ее поведение в той или иной ситуации. Для такого наблюдения исследователь должен располагать соответствующими «смотровыми окнами», которые можно было бы при необходимости закрыть, перенести на другое место, изменить масштаб и форму представления наблюдаемых характеристик и т.д.,

причем не дожидаясь окончания текущего модельного эксперимента. Имитационная модель в этом случае выступает как источник ответа на вопрос: «что будет, если...».

Реализация таких возможностей на универсальном языке программирования – дело очень непростое. В настоящее время существует довольно много программных продуктов, позволяющих моделировать процессы. К таким пакетам относятся: Pilgrim, GPSS, Simplex и ряд других.

Вместе с тем в настоящее время на российском рынке компьютерных технологий есть продукт, позволяющий весьма эффективно решать указанные проблемы, – пакет MATLAB, содержащий в своем составе инструмент визуального моделирования Simulink.

Simulink – это инструмент, позволяющий достаточно быстро смоделировать систему и получить показатели ожидаемого эффекта и сравнить их с затратами сил на их достижение.

Раздел I.

Теоретические основы имитационного моделирования

Глава 1. Основные понятия теории моделирования экономических систем и процессов

§1. Основы принятия решений относительно создания, совершенствования, развития экономических систем

В течение всей своей жизни, от первых шагов и до последнего вздоха, человек вынужден принимать те или иные решения: какой подарок попросить у Деда Мороза, куда пойти учиться, как лучше потратить лишние (или последние) деньги и т.д.

Если определенная ситуация, требующая принятия решения, повторяется достаточно часто, то решение приходит «само собой», автоматически. Если же ситуация недостаточно знакома или человек не располагает всей необходимой информацией, то принятие решения существенно усложняется. В таких случаях он вынужден, как правило, сравнивать между собой несколько возможных вариантов и выбирать тот, который кажется ему наиболее предпочтительным (или наименее опасным).

Значительно более важные последствия имеют так называемые *управляющие решения*. Из теории исследования операций известно, что количественно принятое решение можно оценить с помощью *показателя эффективности* функционирования исследуемой (разрабатываемой) системы, в том числе экономической.

В общем случае он отражает результат функционирования системы (или, говоря терминами теории исследования операций, – проведения операции), который, в свою очередь, является функцией трех факторов: полезного эффекта операции (q), затрат ресурсов на проведение операции (c) и затрат времени на ее проведение (t). Значения величин q , c и t зависят от стратегии проведения операции (u). В формальном виде сказанное можно записать так:

$$Y_{on} = Y(q(u), c(u), t(u)).$$

Пример. Принимается решение относительно строительства крупного складского помещения для дорогостоящего товара (например автомобилей). От того, насколько правильно будет выбран показатель эффективности, определены факторы, зависит выбор правильного решения.

$q(u)$ – число автомобилей, хранящихся на складе;

$c(u)$ – стоимость доставки и хранения автомобилей (в рублях);

$t(u)$ – время доставки и хранения автомобилей (в часах).

Имея возможность рассчитать указанные величины, можно получить достаточно объективную оценку эффективности выбранного способа продажи автомобилей.

Показатель эффективности (ПЭ) позволяет оценить (точнее, описать) результат операции, полученный при использовании конкретной стратегии. Однако даже если такие оценки будут получены для всего множества допустимых стратегий, этого еще недостаточно, чтобы выбрать одну из них, ту, которая будет реализована. Например, при оценке загруженности вычислительной сети оказалось, что коэффициент использования равен 0,7. Хорошо это или плохо? Чтобы ответить на подобный вопрос, необходимо сформулировать правило, позволяющее лицу, принимающему решение (ЛПР), сравнивать между собой стратегии, характеризующиеся различными значениями ПЭ. В одних случаях правило сравнения может быть очень простым, в других же его вообще не удастся найти и приходится изменять (уточнять) показатель эффективности.

Скажем, если автомобиль одной и той же марки в двух разных автосалонах продается по разным ценам (при прочих равных условиях), то правило выбора салона напрашивается само. Со всем другое дело, когда автомобили различаются стоимостью, фирмой-изготовителем, дизайном, организацией гарантийного обслуживания и т.д. При этом первый оказывается предпочтительнее по одним показателям, второй – по другим. В такой ситуации покупатель должен сначала определить правило выбора и только после этого сравнивать между собой различные варианты.

В теории принятия решений правило, на основании которого производится выбор стратегии, отвечающей интересам ЛПР, называется *критерием эффективности*.

Таким образом, показатель эффективности и критерий эффективности в совокупности отражают цели, которые преследует ЛПР при проведении данной операции, а также наиболее предпочтительный для него способ достижения этой цели.

Лица, ответственные за принятие решений, касающихся проектирования и создания экономических систем, могут оценить их эффективность одним из трех следующих способов.

Во-первых, есть возможность (по крайней мере теоретическая) проводить управляемые эксперименты с экономической системой фирмы, отрасли или страны. Однако принятие неоптимальных решений может причинить ущерб экономической системе. При этом чем больше масштаб системы, тем ощутимее убытки. Тем не менее, на практике такие эксперименты нередко производились и производятся (в некоторых странах) с неизменно отрицательным результатом.

Даже в случае оптимальных решений, касающихся, например, управления деятельностью фирмы, при проведении натуральных экспериментов трудно сохранить постоянство факторов и условий, влияющих на результат, а следовательно, сложно обеспечить надежную оценку различных экономических решений.

Во-вторых, если есть данные о развитии экономической системы за некоторый период времени в прошлом, то можно

провести мысленный эксперимент на этих данных. Однако для этого нужно знать точно, какие изменения и каких входных переменных привели к наблюдаемому изменению выходных переменных, характеризующих эффективность экономической системы. Иногда причинами изменений могут оказаться случайные возмущения, или так называемый «шум». Поэтому нельзя слишком доверять оценкам экономических решений, полученным на основе данных о развитии системы в прошлом.

В-третьих, можно построить математическую модель рассматриваемой системы, связывающую входные (независимые) переменные с выходными (зависимыми) переменными, а также с экономической стратегией, т.е. со способом управления экономической системой. Если есть основания для того, чтобы считать разработанную математическую модель *адекватной* рассматриваемой экономической системе, то с помощью модели можно производить расчеты или машинные эксперименты (если модель реализована на ЭВМ). По результатам этих экспериментов можно выработать рекомендации по повышению эффективности существующей или проектируемой экономической системы (рис. 1.1).



Рис. 1.1. Принятие решения об эффективности экономической системы

Условием для разработки модели является наличие так называемой *информационной достаточности*. Это означает, что разработчик должен иметь достаточное представление о том, что является входными и выходными переменными в исследуемой системе и какие факторы оказывают влияние на процесс ее функционирования. Если уровень информационной достаточности невысок, то создать модель, с помощью которой можно получать новые знания об объекте-оригинале, невозможно. Если же уровень информационной достаточности велик, т.е. система уже хорошо изучена, то вопрос о создании модели теряет смысл, так как новых знаний она также не даст.

Следовательно, разрабатывать модель имеет смысл только в том случае, если объект-оригинал еще недостаточно изучен или вообще не существует в природе и только проектируется.

Если объект-оригинал существует, то модель считается *адекватной* ему в том случае, если зависимость выходных переменных от входных параметров в модели и в объекте-оригинале практически совпадает. При упрощении моделей степень адекватности снижается.

Если же объекта-оригинала еще не существует, то модель считается *адекватной* ему, если она с достаточной степенью приближения на уровне понимания моделируемого процесса исследователем отражает закономерности процесса функционирования реальной системы. Залогом адекватности в этом случае является полнота описания моделируемого процесса, т.е. учет всех факторов, поддающихся формализации.

Ярким примером успешного решения задачи моделирования процесса, который невозможно осуществить на практике, является разработка вычислительным центром АН СССР в 1985 г. под руководством академика Н.Н. Моисеева модели ядерной войны, получившей название «Гея». С помощью этой модели были строго научно оценены катастрофические для человека и всего живого на Земле последствия, к которым привела бы ядерная война. Опубликование результатов исследований внесло важный вклад в ослабление ядерной угрозы.

Существует множество различных типов моделей: физические, аналоговые, интуитивные и т.д. Особое место среди них занимают математические модели, которые, по мнению академика А.А. Самарского, «являются самым большим достижением научно-технической революции XX века». Математические модели делятся на две группы: *аналитические* и *алгоритмические* (которые иногда называют *имитационными*).

В настоящее время нельзя назвать область человеческой деятельности, в которой в той или иной степени не использовались бы методы моделирования. Не составляет исключения и экономическая деятельность. Однако в области имитационного моделирования экономических процессов до сих пор наблюдаются некоторые сложности.

На наш взгляд, это обстоятельство объясняется следующими причинами.

1. Экономические процессы происходят в значительной мере стихийно, неуправляемо. Они плохо поддаются попыткам волевого управления со стороны политических, государственных и хозяйственных руководителей отдельных отраслей и экономики страны в целом. По этой причине экономические системы плохо поддаются изучению и формализованному описанию.

2. Специалисты в области экономики, как правило, имеют недостаточную математическую подготовку вообще и по вопросам математического моделирования в частности. Большинство из них не умеет формально описывать (формализовывать) наблюдаемые экономические процессы. Это, в свою очередь, не позволяет установить, адекватна ли та или иная математическая модель рассматриваемой экономической системе.

3. Специалисты в области математического моделирования, не имея в своем распоряжении формализованного описания экономического процесса, не могут создать адекватную ему математическую модель.

Существующие математические модели, которые принято называть *моделями экономических систем*, можно условно разделить на три группы.

К *первой группе* можно отнести модели, достаточно точно отражающие какую-либо одну сторону определенного экономического процесса, происходящего в системе сравнительно малого масштаба. С точки зрения математики они представляют собой весьма простые соотношения между двумя-тремя переменными. Обычно это алгебраические уравнения 2-й или 3-й степени, в крайнем случае система алгебраических уравнений, требующая для решения применения метода итераций (последовательных приближений). Они находят применение на практике, но не представляют интереса с точки зрения специалистов в области математического моделирования.

Ко *второй группе* можно отнести модели, которые описывают реальные процессы, протекающие в экономических системах малого и среднего масштаба, подверженные воздействию случайных и неопределенных факторов. Разработка таких моделей требует принятия допущений, позволяющих разрешить неопределенности. Например, требуется задать распределения случайных величин, относящихся к входным переменным. Эта искусственная операция в известной степени порождает сомнение в достоверности результатов моделирования. Однако другого способа создания математической модели не существует.

Среди моделей этой группы наибольшее распространение получили модели так называемых систем массового обслуживания. Существуют две разновидности этих моделей: аналитические и алгоритмические. Аналитические модели не учитывают действие случайных факторов и поэтому могут использоваться только как модели первого приближения. С помощью алгоритмических моделей исследуемый процесс может быть описан с любой степенью точности на уровне его понимания постановщиком задачи.

К *третьей группе* относятся модели больших и очень больших (макроэкономических) систем: крупных торговых и промышленных предприятий и объединений, отраслей народного хозяйства и экономики страны в целом. Создание ма-

тематической модели экономической системы такого масштаба представляет собой сложную научную проблему, решение которой под силу лишь крупному научно-исследовательскому учреждению.

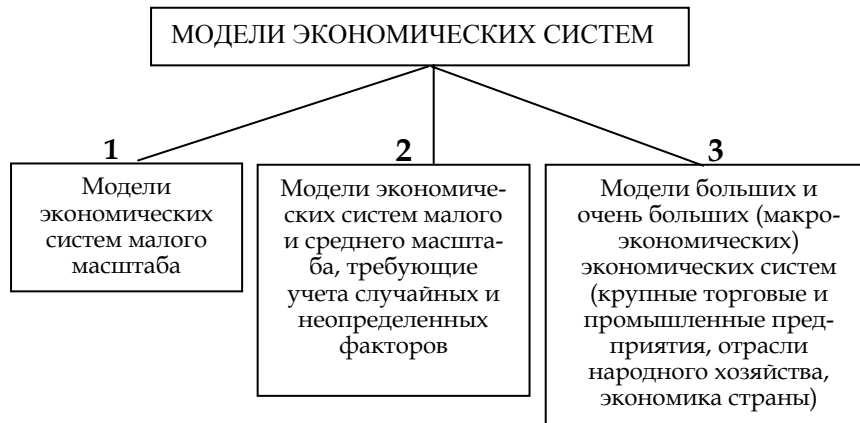


Рис. 1.2. Классификация моделей экономических систем

Экономические модели, рассматриваемые в данной учебной дисциплине, относятся ко второй группе (рис. 1.2).

§2. Основы имитационного моделирования

2.1. Понятие модели

В настоящее время нельзя назвать область человеческой деятельности, в которой в той или иной степени не использовались бы методы моделирования. Между тем общепризнанного определения понятия модели не существует.

На наш взгляд, заслуживает предпочтения следующее определение:

модель – объект любой природы, который создается исследователем с целью получения новых знаний об объекте

оригинале и отражает только существенные (с точки зрения разработчика) свойства оригинала.

Анализируя содержание этого определения, можно сделать следующие выводы:

1) любая модель *субъективна*, она несет на себе печать индивидуальности исследователя;

2) любая модель *гомоморфна*, т.е. в ней отражаются не все, а только существенные свойства объекта-оригинала;

3) возможно существование *множества моделей* одного и того же объекта-оригинала, отличающихся целями исследования и степенью адекватности.

Модель считается *адекватной* объекту-оригиналу, если она с достаточной степенью приближения на уровне понимания моделируемого процесса исследователем отражает закономерности процесса функционирования реальной системы во внешней среде.

2.2. Классификация моделей

По форме представления объектов модели можно разделить на две группы: *материальные и идеальные* (рис. 1.3).

Материальные модели, в свою очередь, делятся на физические и аналоговые. В *физических моделях* обеспечивается аналогия физической природы и модели (примером может служить аэродинамическая труба). В *аналоговых моделях* добиваются сходства процессов, протекающих в оригинале и модели (так, с помощью гидроинтегратора моделируется передача тепла).

Идеальные модели можно разделить на знаковые (семиотические) и интуитивные (мысленные). *Интуитивные* модели используются для прогнозирования на основе анализа наблюдений прошлого периода: объема продаж, прибыли и денежного потока. При этом не предпринимаются попытки объяснить причинные взаимосвязи, которые лежат в основе интуитивной модели.

Знаковые модели делятся на логические, геометрические и математические. *Логические модели* – модели, в которых представлены различные варианты выбора действий на основе умозаключений и анализа условий. *Геометрические* – это графические формы и объемные конструкции. Например: рисунок, пиктограмма, чертеж, карта, план, объемное изображение и т.д.

Математические модели можно разделить на аналитические, алгоритмические (имитационные) и комбинированные.

Для *аналитического* моделирования характерно то, что для описания процессов функционирования системы используются системы алгебраических, дифференциальных, интегральных или конечно-разностных уравнений. Аналитическая модель может быть исследована следующими методами:

- а) аналитическим, когда стремятся получить в общем виде явные зависимости для искомых характеристик;
- б) численным, когда, не умея решать уравнения в общем виде, стремятся получить числовые результаты при конкретных начальных данных;
- в) качественным, когда, не имея решения в явном виде, можно найти некоторые свойства решения (например, оценить устойчивость решения).

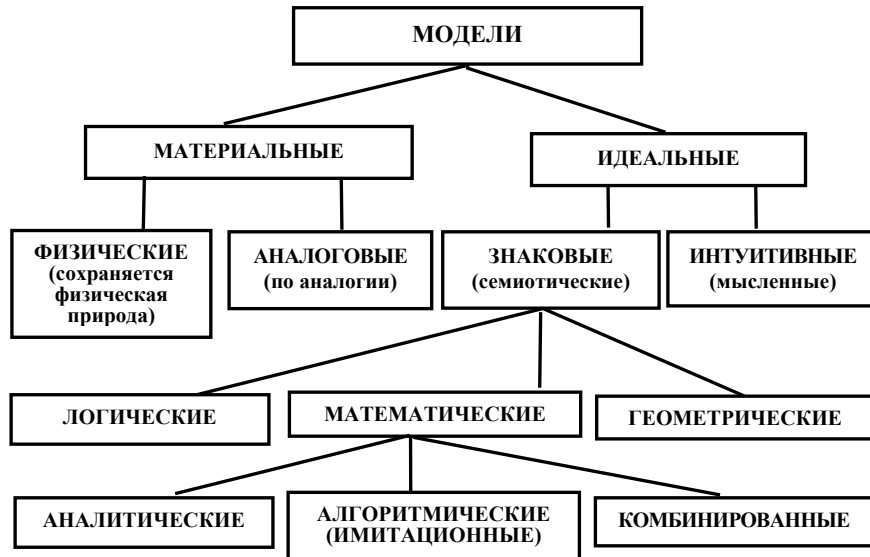


Рис. 1.3. Классификация моделей

Желая использовать аналитический метод, часто идут на существенные упрощения первоначальной модели, чтобы иметь возможность изучить хотя бы общие свойства системы. Аналитические модели бывают *детерминированные* и *статистические*. Численный метод проведения аналитических расчетов с помощью датчиков случайных чисел получил название метода статистических испытаний, или метода Монте-Карло.

При *алгоритмическом (имитационном) моделировании* описывается процесс функционирования системы во времени, причем имитируются элементарные явления, составляющие процесс, с сохранением их логической структуры и последовательности протекания во времени. Имитационные модели также могут быть детерминированными и статистическими. В последнем случае в модели с помощью датчиков случайных чисел имитируется действие неопределенных и случайных

факторов. Такой метод моделирования получил название метода статистического моделирования. В настоящее время этот метод считается наиболее эффективным методом исследования сложных систем, а часто и единственным практически доступным методом получения информации о поведении гипотетической системы на этапе ее проектирования.

Комбинированное моделирование позволяет объединить достоинства аналитического и алгоритмического моделирования. При построении комбинированных моделей производится предварительная декомпозиция процесса функционирования модели на составляющие подпроцессы. Для тех из них, где это возможно, используются аналитические модели, а для остальных процессов строятся алгоритмические модели.

2.3. Последовательность разработки математических моделей

Слово «компьютер» пока в нашем повествовании не использовалось. Тем не менее рано или поздно оно должно было появиться. Начнем со словосочетания «компьютерное моделирование», которое все чаще используется в соответствующей литературе. Само по себе это понятие весьма широкое и каждый автор трактует его по-своему. Встречаются, например, такие выражения: «компьютерное моделирование верхней одежды», «компьютерное моделирование причесок» и т.п. В связи с этим есть необходимость уточнить, что же мы будем понимать под этим термином. В данном случае компьютерное моделирование – это математическое моделирование с использованием средств вычислительной техники. Соответственно, технология компьютерного моделирования предполагает выполнение следующих действий:

- определение цели моделирования;
- построение концептуальной модели;
- разработка алгоритма модели системы, формализация модели;
- разработка программы модели системы;
- планирование модельных экспериментов;

- реализация плана эксперимента (проведение машинных экспериментов с моделью системы);
- анализ и интерпретация результатов моделирования.

2.3.1. Определение цели моделирования

Общая цель моделирования в процессе принятия решения была сформулирована ранее – это определение (расчет) значений выбранного показателя эффективности для различных стратегий проведения операции (или вариантов реализации проектируемой системы). При разработке конкретной модели цель моделирования должна уточняться с учетом используемого критерия эффективности. Таким образом, цель моделирования определяется как целью исследуемой операции, так и планируемым способом использования результатов исследования.

Например, проблемная ситуация, требующая принятия решения, формулируется следующим образом: найти вариант построения вычислительной сети, который обладал бы минимальной стоимостью при соблюдении требований производительности и надежности. В этом случае целью моделирования является отыскание параметров сети, обеспечивающих минимальное значение ПЭ, в роли которого выступает стоимость.

Задача может быть сформулирована иначе: из нескольких вариантов конфигурации вычислительной сети выбрать наиболее надежный. Здесь в качестве ПЭ выбирается один из показателей надежности (средняя наработка на отказ, вероятность безотказной работы и т.д.), а целью моделирования является сравнительная оценка вариантов сети по этому показателю.

Приведенные примеры позволяют напомнить о том, что сам по себе выбор показателя эффективности еще не определяет «архитектуру» будущей модели, поскольку на этом этапе не сформулирована ее концепция, или, как говорят, не определена концептуальная модель исследуемой системы.

2.3.2. Построение концептуальной модели

Концептуальная (содержательная) модель – это абстрактная модель, определяющая структуру моделируемой системы, свойства ее элементов и причинно-следственные связи, присущие системе и существенные для достижения цели моделирования.

Построение концептуальной модели включает следующие этапы:

- определение типа системы;
- описание рабочей нагрузки (определение параметров и переменных модели);
- декомпозицию системы.

На первом этапе осуществляется сбор фактических данных (на основе работы с литературой и технической документацией, проведения натурных экспериментов, сбора экспертной информации и т.д.), а также выдвижение гипотез относительно значений параметров и переменных, для которых отсутствует возможность получения фактических данных. Если полученные результаты соответствуют принципам информационной достаточности и осуществимости, то они могут служить основой для отнесения моделируемой системы к одному из известных типов (классов). Типы систем будут рассмотрены позднее.

Описание рабочей нагрузки. При исследовании эффективности операции (функционирования системы) весьма важную роль играет корректное описание условий ее протекания. Как правило, оно представляет собой перечень и характеристики внешних факторов, воздействующих на исполнительную подсистему, используемую ЛППР для достижения целей операции. Если при сравнении различных стратегий другие виды материальных ресурсов не рассматриваются, то задача исследования эффективности операции может быть сформулирована как задача оценки эффективности исполнительной подсистемы (именно в этом смысле ранее наряду с понятием «эффективность операции» использовалось поня-

тие «эффективность системы»). В этом случае вместо условий проведения операции удобнее рассматривать рабочую нагрузку соответствующей системы.

Рабочая нагрузка – это совокупность внешних воздействий, оказывающих влияние на эффективность применения данной системы в рамках проводимой операции.

Например, пусть оценивается производительность строящейся бензоколонки. В качестве параметров рабочей нагрузки такой системы целесообразно рассматривать поток автомобилей, подлежащих заправке, и поток отказов, приводящий к нарушению ритма работы бензоколонки.

Описание рабочей нагрузки является не только важной, но и достаточно сложной задачей. Особенно в тех случаях, когда приходится учитывать влияние случайных факторов или когда речь идет о рабочей нагрузке принципиально новой проектируемой системы.

Декомпозиция системы производится исходя из выбранного уровня детализации модели, который, в свою очередь, определяется тремя факторами:

- целями моделирования;
- объемом априорной информации о системе;
- требованиями к точности и достоверности результатов моделирования.

Уровни детализации иногда называют стратами, а процесс выделения уровней – стратификацией.

Детализация системы должна производиться до такого уровня, чтобы для каждого элемента были известны или могли быть получены зависимости его выходных характеристик от входных воздействий, существенные с точки зрения выбранного показателя эффективности.

Повышение уровня детализации описания системы позволяет получить более точную ее модель, но усложняет процесс моделирования и ведет к росту затрат времени на его проведение.

Например, если моделируется дискретная система, то более детальное ее описание означает увеличение числа раз-

личных состояний системы, учитываемых в модели, и, как следствие, – неизбежный рост объема вычислений.

Поэтому при выборе уровня описания системы целесообразно руководствоваться следующим правилом: в модель должны войти все параметры, которые обеспечивают определение интересующих исследователя характеристик системы на заданном временном интервале ее функционирования; остальные параметры по возможности из модели следует исключить.

При имитационном моделировании для оценки выбранного уровня детализации можно использовать специальные критерии.

Первый из них – отношение реального времени функционирования системы к времени моделирования (т.е. к затратам машинного времени, необходимого на проведение модельного эксперимента). Например, если при одних и тех же подходах к программной реализации модели моделирование одного часа работы системы требует в одном случае 3 минуты машинного времени, а в другом – 10 минут, то во втором случае степень детализации описания выше (соотношение 3:10).

Второй критерий – разрешающая способность модели, в том числе:

- разрешающая способность *по времени* – может быть определена как кратчайший интервал модельного времени между соседними событиями;
- разрешающая способность *по информации* – наименьшая идентифицируемая порция информации, представимая в модели (для вычислительных систем, например, такими порциями могут быть: слово, страница, программа, задание).

Третий критерий – число различных моделируемых состояний системы (или типов событий). Для тех компонентов, относительно которых известно или предполагается, что они сильнее влияют на точность результатов, степень детальности может быть выше других. Необходимо отметить, что с увеличением детальности возрастает устойчивость модели, но возрастают и затраты машинного времени на проведение модельного эксперимента.

Разработка концептуальной модели завершается составлением содержательного описания, которое используется как основной документ, характеризующий результаты работы на первом этапе.

2.3.3. Разработка алгоритма модели системы

Разработка алгоритма модели включает следующие подэтапы:

- построение логической схемы алгоритма;
- получение математических соотношений;
- проверку достоверности алгоритма.

Вначале создается укрупненная (обобщенная) схема моделирующего алгоритма, которая задает общий порядок действий при моделировании исследуемого процесса. Затем разрабатывается детальная схема, каждый элемент которой впоследствии превращается в оператор программы.

Для комбинированных моделей разрабатывается аналитическая часть в виде явных функций и имитационная часть в виде моделирующего алгоритма.

Проверка достоверности алгоритма должна дать ответ на вопрос, насколько алгоритм отражает замысел моделирования, сформулированный на этапе разработки концептуальной модели.

2.3.4. Разработка программы модели системы

Разработка программы для ЭВМ включает следующие подэтапы:

- выбор вычислительных средств;
- проведение программирования;
- проверку достоверности программы.

Прежде всего выбираются тип ЭВМ (компьютера) и язык программирования или программа моделирования.

После составления программы производится проверка ее достоверности на контрольном примере. На этом подэтапе необходимо оценить затраты машинного времени для расчета одной реализации моделируемого процесса, что позволит

разработчику модели правильно сформулировать требования к точности и достоверности результатов моделирования.

2.3.5. Планирование модельных экспериментов и проведение машинных экспериментов с моделью системы

На этом этапе проводятся серийные расчеты по составленной и отлаженной программе. Этап включает следующие подэтапы:

- планирование машинного эксперимента;
- проведение рабочих расчетов;
- представление результатов моделирования;
- интерпретацию результатов моделирования;
- выдачу рекомендаций по оптимизации режима работы

реальной системы.

Перед проведением рабочих расчетов на ЭВМ должен быть составлен план проведения эксперимента с указанием комбинаций переменных и параметров, для которых должно проводиться моделирование системы. Задача заключается в разработке оптимального плана эксперимента, реализация которого позволяет при сравнительно небольшом числе испытаний модели получить достоверные данные о закономерностях функционирования системы.

Результаты моделирования могут быть представлены в виде таблиц, графиков, диаграмм, схем и т.п. В большинстве случаев наиболее простой формой считаются таблицы, хотя графики более наглядно иллюстрируют результаты моделирования системы.

Целесообразно предусмотреть вывод результатов на экран дисплея и на принтер.

Интерпретация результатов моделирования имеет целью переход от информации, полученной в результате машинного эксперимента с моделью, к выводам, касающимся процесса функционирования объекта-оригинала.

На основании анализа результатов моделирования принимается решение о том, при каких условиях система будет функционировать с наибольшей эффективностью.

Глава 2. Математические схемы моделирования экономических систем

§1. Классификация моделируемых систем

В процессе создания математической модели, реализуемой на ЭВМ, происходит переход от содержательного описания к формальному алгоритму. Промежуточным звеном между ними может служить *математическая схема*.

Существует ряд типовых математических схем, которые могут лечь в основу разрабатываемого конкретного моделирующего алгоритма.

Выбор типа математической схемы зависит от моделируемой системы. Системы по типу поведения классифицируются следующим образом (рис. 2.1).

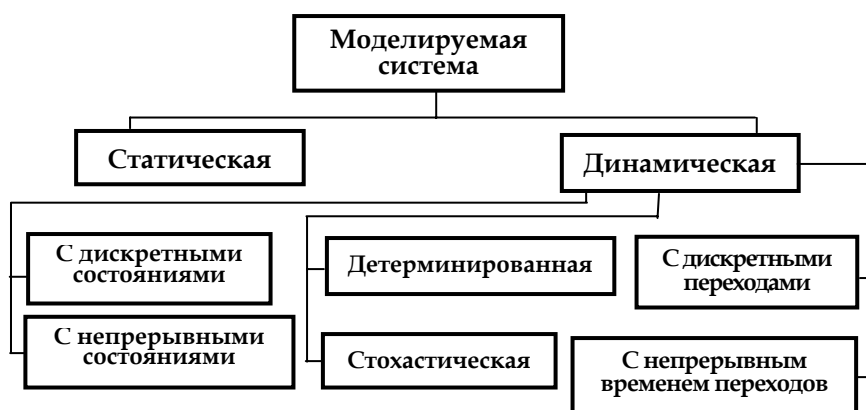


Рис. 2.1. Классификация систем по типу поведения

Одним из них является мощность множества состояний моделируемой системы. По этому признаку системы делят на статические и динамические. Система называется статической, если множество ее состояний содержит один элемент. Если состояний больше одного и они могут изменяться во времени,

система называется динамической. Процесс смены состояний называется движением системы:

Различают следующие типы динамических систем.

- с дискретными состояниями (множество состояний конечно, или счетно);

- с непрерывным множеством состояний.

Системы с дискретными состояниями характеризуются тем, что в любой момент времени можно однозначно определить, в каком именно состоянии находится система. Для такой идентификации обязательно нужно знать тот признак, который отличает одно состояние системы от другого. Например, при исследовании систем массового обслуживания в качестве такого признака обычно используют число заявок в системе. Соответственно, изменение числа заявок в системе интерпретируется как переход системы в новое состояние.

Если же не удастся подобрать такой признак либо его текущее значение невозможно зафиксировать, то систему относят к классу систем с непрерывным множеством состояний. На практике возможны также смешанные случаи, когда некоторые состояния системы могут быть идентифицированы как дискретные, а другие – как непрерывные.

Смена состояний может происходить либо в фиксированные моменты времени, множество которых дискретно (например, поступление новых заявок на обслуживание), либо непрерывно. В соответствии с этим различают системы с дискретным временем переходов (смены состояний) и системы с непрерывным временем (точнее, «живущие» в непрерывном времени).

По условиям перехода из одного состояния в другое различают детерминированные системы и стохастические.

В *детерминированных системах* новое состояние зависит только от времени и текущего состояния системы. Другими словами, если имеются условия, определяющие переход системы в новое состояние, то для детерминированной систе-

мы можно однозначно указать, в какое именно состояние она перейдет.

Для *стохастической системы* можно указать лишь множество возможных состояний перехода и, в некоторых случаях, – вероятностные характеристики перехода в каждое из этих состояний.

Рассмотренная схема классификации систем важна не сама по себе. На этапе разработки концептуальной модели она, во-первых, позволяет уточнить цели и задачи моделирования и, во-вторых, облегчает переход к этапу формализации модели. Кроме того, значительно позже, на этапе оценки качества разработанной модели, знание классификационных признаков дает возможность оценить степень ее соответствия первоначальному замыслу разработчика.

Необходимо отметить, что рассмотренные классификационные признаки применимы и для определения типа разрабатываемой модели. При этом исследуемая система и ее модель могут относиться как к одному, так и к разным классам. Например, реальная система может быть подвержена воздействию случайных факторов и, соответственно, будет относиться к классу стохастических систем. Если разработчик модели считает, что влиянием этих факторов можно пренебречь, то создаваемая модель будет представлять собой детерминированную систему. Аналогичным образом возможно отображение системы с непрерывным временем смены состояний в модель с дискретными переходами и т.д. Разумеется, принадлежность реальной системы и ее модели к одному классу говорит о корректности модели, однако с точки зрения интересов исследования такое «зеркальное отображение» далеко не всегда является полезным (вспомните принцип множественности моделей). Исходя из типов поведения систем подбирается та или иная математическая схема (модель).

§2. Математические схемы (модели)

Существуют следующие математические схемы (модели):

- непрерывно-детерминированные (D-схемы);
- дискретно-детерминированные (F-схемы);
- дискретно-стохастические (P-схемы);
- непрерывно-стохастические (Q-схемы).

К *непрерывно-детерминированным моделям* относятся модели, описываемые системами обыкновенных дифференциальных уравнений или уравнений в частных производных. В качестве независимой переменной, от которой зависят неизвестные искомые функции, обычно служит время. Тогда вектор-функция искомых переменных будет непрерывной. Математические схемы такого вида отражают динамику изучаемой системы и поэтому называются D-схемами (англ. dynamic).

К *дискретно-детерминированным моделям* относятся так называемые конечные автоматы. Автомат можно представить как некоторое устройство, на которое подаются входные сигналы и снимаются выходные и которое может иметь некоторые внутренние состояния. У конечного автомата множество входных сигналов и внутренних состояний является конечным множеством. Название F-схема происходит от англ. finite automata.

К *дискретно-стохастическим моделям* относятся вероятностные (стохастические) автоматы, или по-английски probabilistic automat. Отсюда название P-схема. В общем виде вероятностный автомат можно определить как дискретный потактный преобразователь информации с памятью, функционирование которого в каждом такте зависит только от состояния памяти в нем и может быть описано стохастически.

Примером типовой схемы *непрерывно-стохастического типа* может служить схема *системы массового обслуживания* (СМО), или по-английски queueing system. Отсюда название Q-схема.

В качестве процесса обслуживания в СМО могут быть представлены различные по физической природе процессы функционирования экономических, производственных, технических и других систем, например потоки товаров, потоки продукции, потоки деталей, потоки клиентов и т.п. Для любой системы массового обслуживания (рис. 2.2) характерно наличие трех отличительных свойств:

- объектов, у которых может возникнуть потребность в удовлетворении некоторых заявок;
- агрегатов, предназначенных для удовлетворения заявок на обслуживание;
- специальной организации приема в систему заявок и их обслуживания.

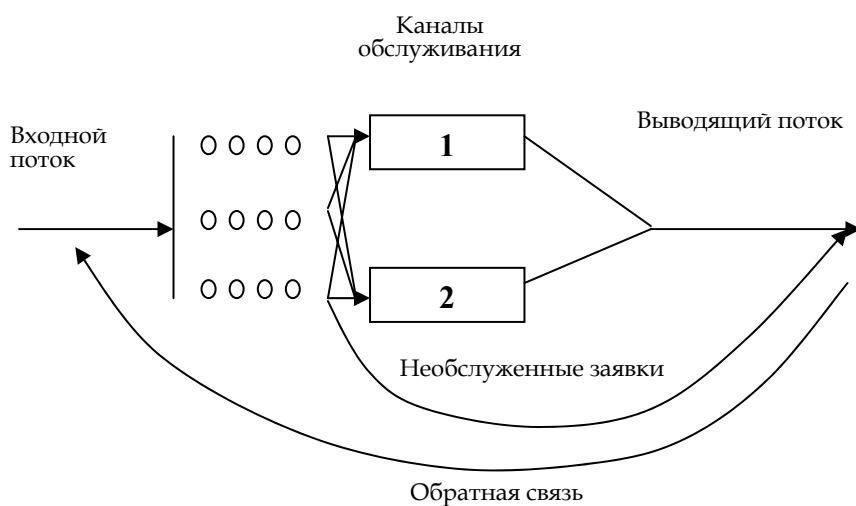


Рис. 2.2. Схема системы массового обслуживания

Совокупность заявок рассматривают как *поток событий*, т.е. последовательность событий, происходящих в случайные моменты времени. Время обслуживания заявки также считается случайной величиной.

Из-за совместного действия этих двух случайных факторов количество обслуженных заявок в заданном интервале времени является величиной случайной. Так как поток заявок и время обслуживания случайные величины, значит, количество заявок, обслуженных в заданном интервале времени, – случайная величина (рис. 2.3).



Рис. 2.3. Временная диаграмма СМО

Исследование моделей СМО ставит целью установление параметров случайных величин, характеризующих процесс обслуживания заявок.

Существует несколько разновидностей СМО:

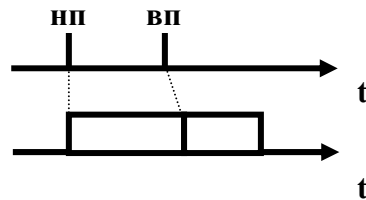
- 1) по числу каналов обслуживания СМО делятся на *одноканальные и многоканальные*;
- 2) по числу фаз (последовательно соединенных агрегатов) – на *однофазные и многофазные*;
- 3) по наличию обратной связи – на **разомкнутые** (с бесконечным числом заявок) и **замкнутые** (с конечным числом заявок);
- 4) по наличию очереди – на системы **без очередей** (с потерями заявок), системы с **неограниченным ожиданием** (по времени или длине очереди) и системы с **ограниченным ожиданием** (по времени или длине очереди);

5) по принципу формирования очередей – на системы с **общей очередью** и системы с **несколькими очередями**;

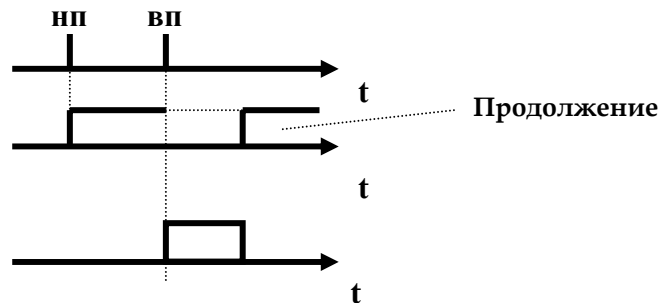
6) по наличию отказов – на системы с **отказами** и системы **без отказов**;

7) по виду приоритета – на системы **со статическим приоритетом** (обслуживание в порядке поступления заявок) и системы с **динамическим приоритетом**, который, в свою очередь, имеет три разновидности:

- **относительный приоритет** (заявка высокого приоритета ожидает окончания обслуживания заявки с более низким приоритетом);



- **абсолютный приоритет** (заявка высокого приоритета при поступлении немедленно вытесняет заявку с более низким приоритетом);



- **смешанный приоритет** (если заявка с низким приоритетом обслуживалась в течение времени, меньше критического, то используется абсолютный приоритет, в противном случае – относительный приоритет).

Глава 3. Моделирование случайных событий и величин

Имитационное моделирование – это численный метод исследования систем и процессов с помощью моделирующего алгоритма.

Каждый раз, когда на ход моделируемого процесса оказывает влияние случайный фактор, его действие имитируется с помощью специально организованного розыгрыша (жеребья). Таким образом строится одна случайная реализация моделируемого явления, представляющая собой как бы один результат опыта. По одному опыту, конечно, нельзя судить о закономерностях изучаемого процесса. Но при большом числе реализации средние характеристики, вырабатываемые моделью, приобретают свойство устойчивости, которое усиливается с увеличением числа реализации.

Бросание жребия можно осуществить вручную (выбором из таблицы случайных чисел), но удобнее это делать с помощью специальных программ, входящих в состав программного обеспечения ЭВМ. Такие программы называют *датчиками*, или *генераторами*, случайных чисел.

В программе Simulink/Matlab имеются стандартные процедуры или функции, которые генерируют случайные (точнее, псевдослучайные) величины с равномерным распределением.

В Simulink/Matlab имеются очень широкие возможности по моделированию различного рода случайностей (событий, процессов, функций и т.д.) (рис. 3.1 и 3.2).

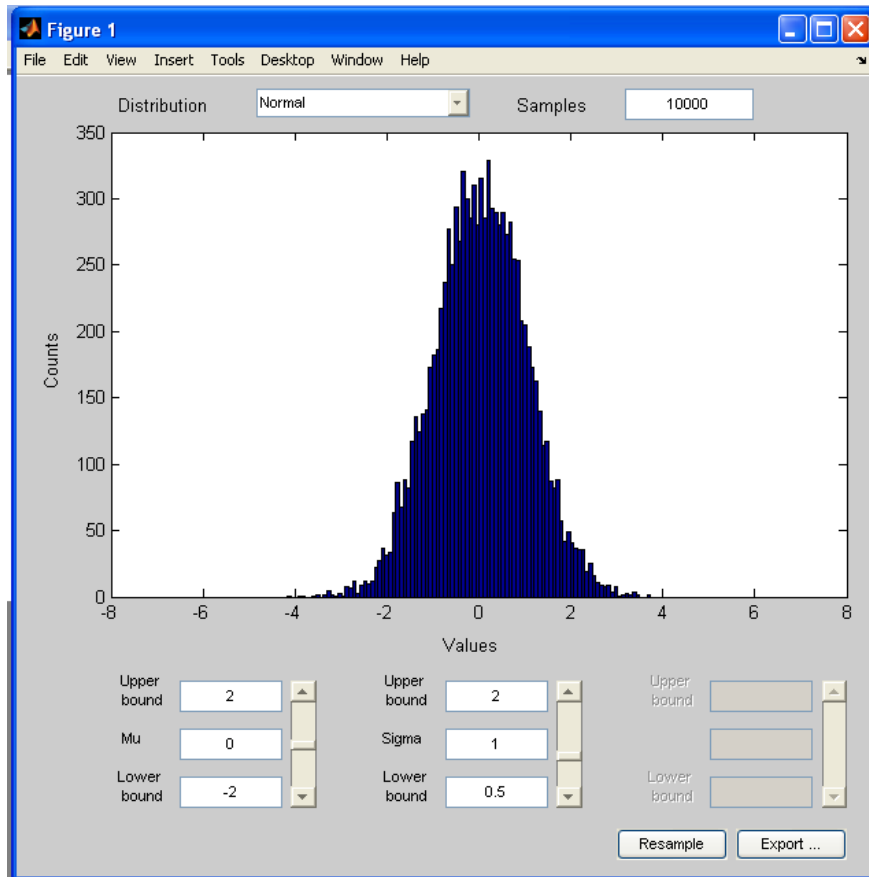
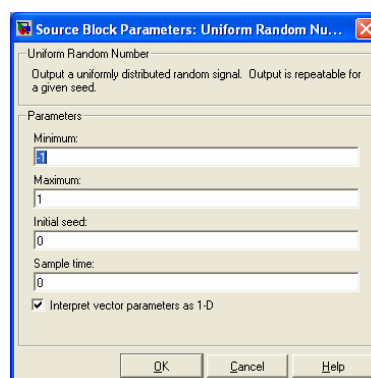
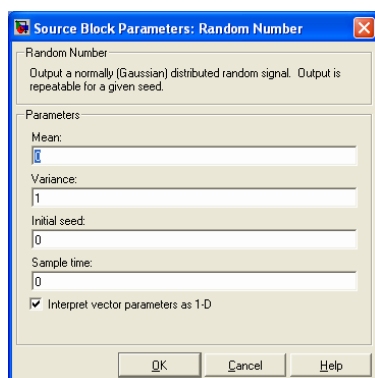
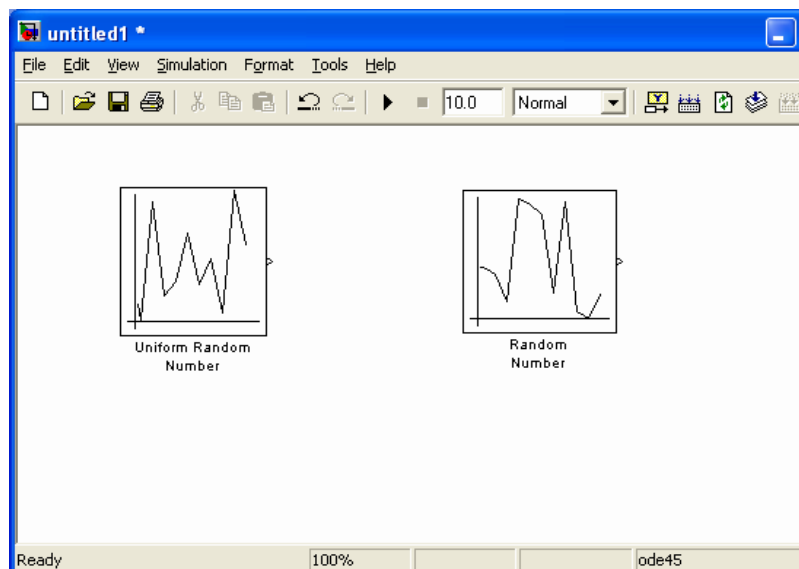


Рис. 3.1. Возможности Matlab по моделированию случайных величин и процессов (раздел Toolboxes-Statistics-Random Number Generation)



Параметры блоков

Рис. 3.2. Возможности Simulink по моделированию случайных процессов

При моделировании экономических процессов возникает необходимость в моделировании различных случайных факторов. Эти факторы в зависимости от их природы могут быть отражены в модели как случайные события, случайные величины (дискретные или непрерывные) или как случайные функции (процессы).

Например, если с помощью создаваемой имитационной модели предполагается исследовать надежность вычислительной системы, то возникновение отказа будет представлено в модели как случайное событие. Если же модель предназначена для оценки временных параметров процесса обслуживания клиентов в автомастерской, то интервал времени до появления очередного клиента удобнее всего описать как случайную величину, распределенную по некоторому закону.

§1. Моделирование случайных событий

1.1. Моделирования простого события

Пусть имеется событие A , вероятность наступления которого равна P_A . Требуется выработать правило, при многократном использовании которого частота появления события стремилась бы к его вероятности. Выберем с помощью датчика случайных чисел, равномерно распределенных в интервале $(0,1)$, некоторое число z и определим вероятность того, что

$$P(z < P_A) = \int_0^{P_A} f(x)dx = P_A.$$

$z < P_A$. Для случайной величины z с равномерным распределением справедлива следующая зависимость:

Таким образом, вероятность попадания случайной величины в интервал $(0, P_A)$ равна величине P_A . Поэтому если при розыгрыше число z попало в этот интервал, то следует считать, что событие A произошло. Противоположное событие (**не** A) произойдет с вероятностью $(1 - P_A)$ в том случае, если $z \geq P_A$.

Процедура моделирования простого события в имитационной модели описывается алгоритмом, схема которого показана на рис. 3.3.

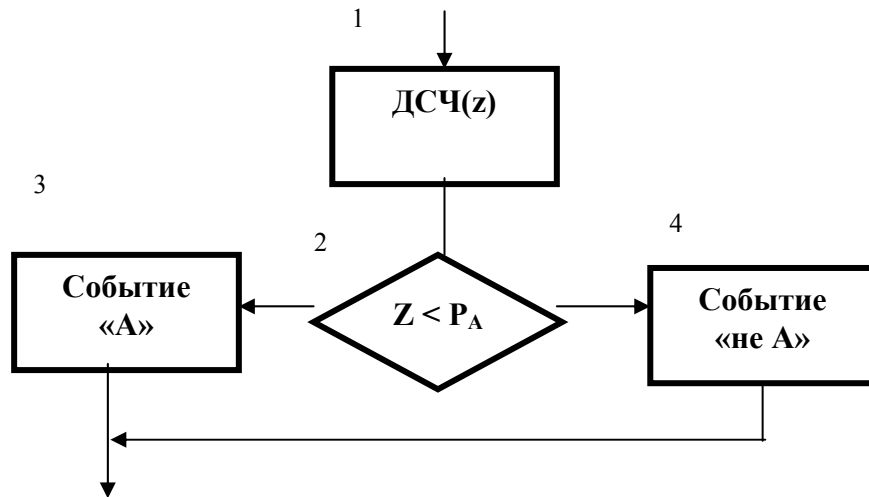


Рис. 3.3. Моделирование простого события

Оператор 1 обращается к датчику случайных чисел, генерирующему случайную величину z . Оператор 2 проверяет условие $z < P_A$. Если оно выполняется, считается, что произошло событие A . В противном случае считается, что произошло противоположное событие (не A).

На рис. 3.4 показан пример реализации модели случайного события с помощью Simulink.

Блок Uniform Random Number генерирует случайные числа (СЧ), равномерно распределенные на интервале $[0;1]$. Для этого выполнены следующие исходные установки его параметров:

- Minimum (нижняя граница диапазона): 0;
- Maximum (верхняя граница диапазона) : 1.

Третий параметр - Sample time - в данном случае позволяет задавать количество СЧ, которые будут сформированы

блоком в течение интервала моделирования: при Sample time, равном 0, новое случайное число генерируется через каждые 0,02 единицы модельного времени; при Sample time, равном интервалу моделирования, генерируется только одно СЧ. На рис. 3.4 приведен пример блок-диаграммы, позволяющей моделировать появление случайного события А при $P_A = 0,4$.

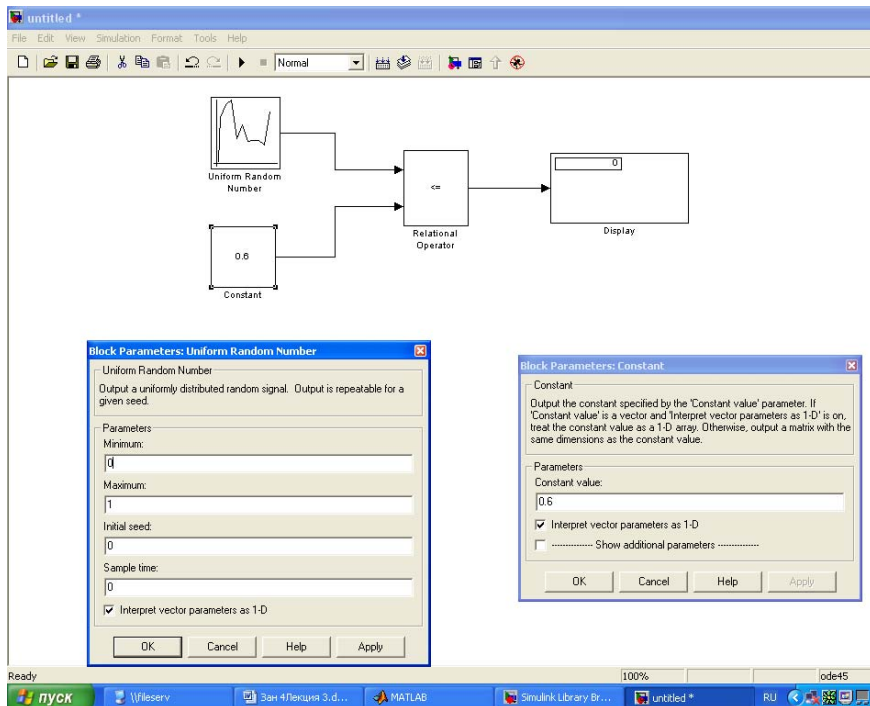


Рис. 3.4. Моделирование случайного события с заданной вероятностью наступления

На рис. 3.5 показано моделирование нескольких случайных событий на интервале моделирования 6 единиц модельного времени.

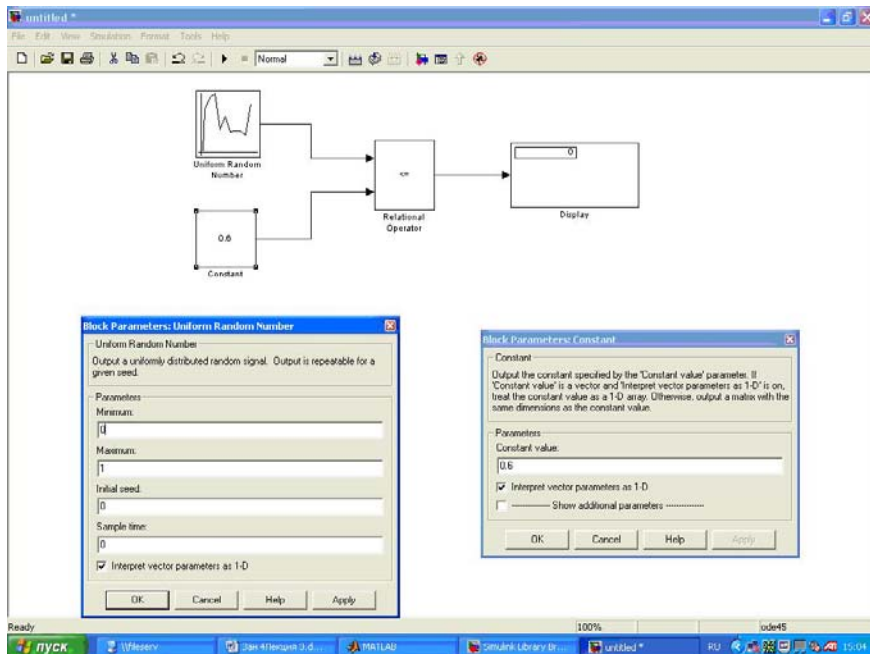


Рис. 3.5. Моделирование ряда случайных событий

Особенностью блока Uniform Random Number является то, что он в каждом сеансе моделирования генерирует одну и ту же последовательность СЧ. Для изменения генерируемой последовательности необходимо вручную изменить значение его параметра Initial seed. При проведении большого числа повторных экспериментов с целью накопления статистических данных это не очень удобно. Поэтому для моделирования случайных событий можно воспользоваться генераторами СЧ, входящими в состав компоненты Matlab, которая называется Toolboxes-Statistics (средства статистического анализа). Данная компонента доступна для использования в том случае, если она включена в рабочую конфигурацию пакета Matlab. Toolboxes-Statistics, как и другие инструментальные приложения, представляет собой набор специализированных функ-

ций (см. рис. 3.1), реализованных в виде М-файлов. Ее особенностью является то, что для нее отсутствует набор блоков, который включался бы в библиотеку Simulink. Поэтому в процессе моделирования статистические функции следует использовать один из двух способов:

- выполнять в командном окне Matlab;
- включать в вычисляемое выражение в тех блоках S-модели, в которых разрешено использование М-функций.

Категория функций Random Number Generation (генераторы случайных чисел) обеспечивает формирование значений случайной величины, распределенной по определенному закону с задаваемыми параметрами. Генератор непрерывной СВ, равномерно распределенной в заданном интервале, называется *unifrnd*. Обращение к данной функции имеет вид *unifrnd* (A, B, M, N), где A, B – границы диапазона распределения, а параметры M, N задают размер генерируемой матрицы случайных чисел. Если параметры M, N опущены, то генерируется единственное значение случайной величины.

При моделировании случайного события функция *unifrnd* может быть указана в качестве параметра настройки следующих блоков:

- Matlab Fcn (раздел Function&Tables);
- Fcn (из того же раздела);
- Constant (раздел Sources).

Напоминание из теории вероятностей

Случайная величина (СВ) – величина, которая в результате опыта может принимать некоторое неизвестное заранее значение.

Дискретная случайная величина (ДСВ) – принимает конечное (счетное) множество возможных значений.

Непрерывная случайная величина (НСВ) – может принимать любые значения из некоторого интервала.

Случайная величина задается **функцией распределения** $F(x) = P(X < x)$. Если $F(x)$ непрерывна и дифференцируема, то непрерывная случайная величина задается **плотностью вероятностей** $f(x)$, которая является производной от $F(x)$.

$$F(x) = \int_{-\infty}^x f(x)dx$$

$$f(x) = F'(x).$$

Свойства функции распределения

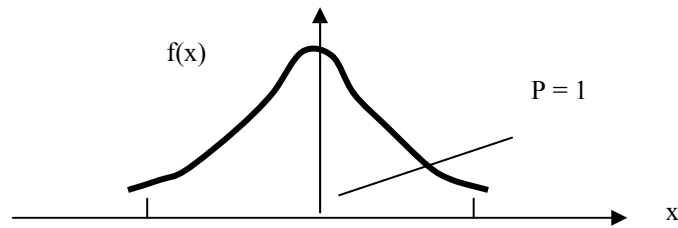
1. При $-\infty < x < +\infty$, $0 < F(x) < 1$;
2. Неубывающая, т.е при $x_2 < x_1$ $F(x_2) > F(x_1)$;
3. Имеет место $F(+\infty) = 1$ и $F(-\infty) = 0$;
4. Вероятность попадания СВ (x) в интервал (a,b).

Свойства плотности распределения вероятностей

1. Неотрицательна $f(x) \geq 0$

$$P(a \leq x \leq b) = F(b) - F(a)$$

2. $\int_{-\infty}^{\infty} f(x)dx = F(\infty) - F(-\infty) = 1$



3. Вероятность попадания в интервал (a,b)

$$P(a \leq x \leq b) = \int_{-\infty}^{\infty} f(x)dx$$

4. Математическое ожидание непрерывной случайной величины

$$M(X) = \int_{-\infty}^{\infty} x \cdot f(x)dx$$

5. Дисперсия непрерывной случайной величины

$$D(X) = M[x - M(X)]^2 = \int_{-\infty}^{\infty} [x - M(X)]^2 \cdot f(x)dx$$

6. Среднее квадратическое отклонение

$$\sigma(X) = +\sqrt{D(X)}.$$

1.2. Моделирование полной группы несовместных событий

Пусть имеется полная группа несовместных событий (ПГНС) A_1, A_2, \dots, A_k с вероятностями P_1, P_2, \dots, P_k . При этом выполняется условие

$$\sum_{i=1}^k P_i = 1.$$

Разделим интервал $(0,1)$ на k отрезков, длины которых составляют P_1, P_2, \dots, P_k (рис. 3.6).

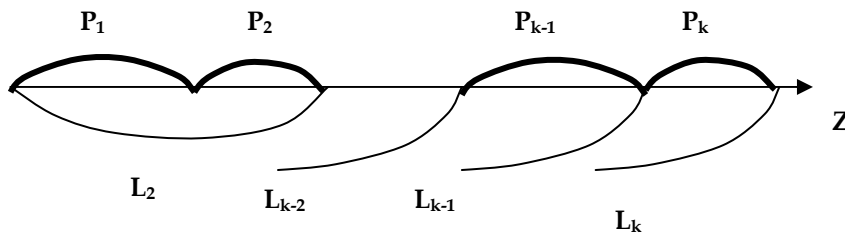


Рис. 3.6. Моделирование полной группы несовместных событий

Если случайное число z , генерированное датчиком случайных чисел с равномерным распределением в интервале $(0,1)$, попало, например, на участок P_{k-1} , то это должно означать, что произошло событие A_{k-1} .

Действительно, если обозначить

$$L_j = \sum_{i=1}^j P_i,$$

то окажется справедливым выражение

$$P(L_{k-2} < z < L_{k-1}) = \int_{L_{k-2}}^{L_{k-1}} 1 \cdot dx = P_{k-1}$$

Следовательно, произойдет событие, которое имеет вероятность P_{k-1} .

Процедура моделирования полной группы несовместных событий описывается алгоритмом, схема которого показана на рис. 3.7.

Оператор 1 обращается к датчику случайных чисел с равномерным распределением в интервале (0,1). Условный оператор 2 проверяет условие попадания случайной величины z в интервал $(0, L_1)$. Если это условие выполняется, то считается, что произошло событие A_1 . Если условие в операторе 2 не выполняется, то алгоритм осуществляет проверку условий попадания случайной величины в другие интервалы. Одно из событий A_1, A_2, \dots, A_k обязательно произойдет.

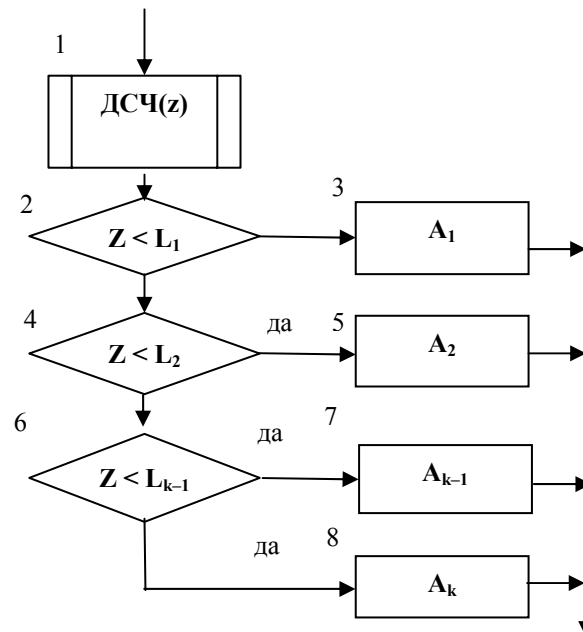


Рис. 3.7. Схема алгоритма моделирования ПНС

§2. Моделирование случайных величин

Использование случайных величин является наиболее универсальным и поэтому наиболее распространенным способом учета в модели случайных факторов, присущих реальным экономическим системам или процессам. Примерами случайных величин могут служить: интервал времени до появления очередного клиента, длительность проведения технического обслуживания автомобиля, объем данных, считываемых из оперативной памяти ЭВМ и т.д. Случайные величины могут быть дискретные или непрерывные. Рассмотрим моделирование тех и других величин.

2.1. Моделирование дискретной случайной величины

Дискретная случайная величина может быть задана табличной зависимостью:

X	x_1	x_2	...	x_n
P	p_1	p_2	...	p_n

Здесь p_j - вероятность того, что дискретная случайная величина X примет значение x_j . При этом $p_1 + p_2 + \dots + p_n = 1$. Разделим интервал $(0,1)$ на n отрезков, длины которых пропорциональны заданным вероятностям. Если случайное число z , вырабатываемое датчиком случайных чисел, равномерно распределенных в интервале $(0,1)$, попадет в интервал p_k , то случайная величина X примет значение x_k . Таким образом, при моделировании дискретных случайных величин фактически используется та же процедура, что и при моделировании ПГНС.

2.2. Моделирование непрерывных случайных величин

При разработке имитационных моделей с использованием универсальных языков программирования используются разработанные методы моделирования различных законов распределения СВ. Если закон СВ распределения известен, то она может быть достаточно адекватно представлена в имитационной модели. Ниже в качестве дополнительного материала представлены методы моделирования СВ, которые могут быть использованы при создании имитационных моделей на основе универсальных языков программирования.

2.2.1. Метод обратной функции

Пусть имеется некоторая непрерывная случайная величина x , заданная функцией распределения $F(x)$. Можно доказать, что значения этой функции равномерно распределены в интервале $(0,1)$. Поэтому между случайной величиной z , равномерно распределенной в том же интервале, и функцией распределения случайной величины x существует взаимно однозначное соответствие, т.е.

$$z = F(x). \quad (1)$$

Отсюда следует, что

$$x = F^{-1}(z), \quad (2)$$

где F^{-1} – обратная функция.

Следовательно, если уравнение (1) имеет аналитическое решение, то для моделирования случайной величины x можно использовать датчик случайных чисел, генерирующий величину z , и затем осуществить расчет по формуле (2).

2.2.2. Моделирование случайных величин с показательным распределением

Пусть имеется случайная величина x с показательным распределением. Функция распределения имеет вид

$$F(x) = 1 - e^{-\lambda x},$$

где λ – параметр распределения.

Применив метод обратной функции, получим

$$z = F(x) = 1 - e^{-\lambda x},$$

откуда

$$x = -\frac{1}{\lambda} \ln(1 - z). \quad (3)$$

Учитывая, что случайная величина $(1 - z)$ имеет также равномерное распределение в интервале $(0,1)$, соотношение (3) можно заменить соотношением

$$x = -\frac{1}{\lambda} \ln(z).$$

2.2.3. Моделирование случайных величин с равномерным распределением

Датчик случайных чисел генерирует случайные величины с равномерным распределением в интервале $(0,1)$. Если же нужно моделировать случайные величины с равномерным распределением в интервале (a,b) , то можно воспользоваться методом обратной функции.

Для рассматриваемого случая выражение (1) примет вид

$$z = F(x) = \frac{x - a}{b - a},$$

откуда $x = a + z(b - a)$.

На практике применяется и другой способ задания равномерного распределения. Вместо границ интервала задаются среднее значение случайной величины x_{cp} и величина интервала Δx . Тогда определение возможного значения случайной величины с равномерным распределением может быть произведено по формуле

$$X = x_{\text{cp}} + \Delta x(z - 0,5).$$

2.2.4. Моделирование случайных величин с нормальным распределением

Нормальное распределение НСВ - это распределение, которое имеет плотность распределения вероятностей

$$f(y) = \frac{1}{\sigma_y \sqrt{2\pi}} \exp \left[-\frac{(y - m_y)^2}{2\sigma_y^2} \right],$$

где m_y - математическое ожидание;

σ_y - среднее квадратическое отклонение.

Функция распределения

$$F(y) = \frac{1}{\sigma_y \sqrt{2\pi}} \int_{-\infty}^y e^{-\frac{(y-m_y)^2}{2\sigma_y^2}} dy.$$

Метод обратной функции для нормального распределения неприменим, так как после подстановки соответствующей функции распределения выражение (2) не имеет аналитического решения. Поэтому в данном случае применяется другой метод.

Согласно центральной предельной теореме теории вероятностей при сложении достаточно большого числа одинаково распределенных независимых случайных чисел получается случайная величина, имеющая нормальное распределение.

Напоминание

Центральная предельная теорема теории вероятностей (теорема Ляпунова) гласит: если СВ x_1, x_2, \dots, x_n независимы, одинаково распределены и имеют конечные математическое ожидание и дисперсию, то распределение суммы этих СВ при увеличении n приближается к нормальному (теорема применима при $n > 10$).

Как показали исследования, уже при сложении более десяти случайных величин с равномерным распределением в интервале (0,1) получается случайная величина, которая с точностью, достаточной для большинства практических задач, может считаться распределенной нормально.

Процедура розыгрыша нормально распределенной случайной величины заключается в следующем.

$$v = \sum_{i=1}^{12} z_i.$$

1. Сложим 12 случайных величин с равномерным распределением в интервале $(0,1)$, т.е. составим сумму

Используя известные теоремы о сумме математических ожиданий и дисперсий независимых случайных величин, можно установить, что в данном случае случайная величина v имеет следующие характеристики:

математическое ожидание

$$M(V) = \sum_{i=1}^{12} M(z) = 12 \cdot \frac{1}{2} = 6;$$

дисперсия:

$$D(V) = \sum_{i=1}^{12} D(z) = 12 \cdot \frac{1}{12} = 1;$$

среднее квадратическое отклонение

$$\sigma(V) = +\sqrt{D(V)} = 1$$

2. Нормируем и центрируем случайную величину v , т.е. перейдем к величине

$$\eta = [v - M(V)] / \sigma(V) = v - 6.$$

3. От нормированной и центрированной величины η перейдем к случайной величине y с заданными параметрами $M(Y)$ и $\sigma(Y)$ по формуле

где $M(Y)$ – известное математическое ожидание случайной величины y ;

$$y = M(Y) + \sigma(Y) \cdot \eta.$$

$\sigma(Y)$ – известное квадратическое отклонение случайной величины y .

2.2.5. Моделирование случайных величин с усеченным нормальным распределением

Усеченное нормальное распределение случайной величины x задается четырьмя параметрами: математическим ожиданием $M(X)$, средним квадратическим отклонением $\sigma(X)$, а также минимальным и максимальным значениями x_1 и x_2 (точками усечения).

Функции распределения случайной величины x определяются равенством

$$F(x) = \begin{cases} 0 & \text{при } x < x_1 \\ [\Phi_0(t) - \Phi_0(t_1)] \cdot A & \text{при } x_1 < x \leq x_2, \\ 1 & \text{при } x > x_2 \end{cases}$$

где

Функция Лапласа Φ_0 – это функция распределения нормированной и центрированной случайной величины t . Эта функция табулирована.

$$t = \frac{x - M(X)}{\sigma(X)}, \quad t_1 = \frac{x_1 - M(X)}{\sigma(X)}, \quad t_2 = \frac{x_2 - M(X)}{\sigma(X)},$$
$$A = \frac{1}{\Phi_0(t_2) - \Phi_0(t_1)};$$

$$\Phi_0(t) = \frac{1}{\sqrt{2\pi}} \int_0^t e^{-\frac{t^2}{2}} dt$$
$$t = \frac{y - M_y}{\sigma_y}.$$

Существуют также формулы для расчета математического ожидания, дисперсии и среднего квадратического отклонения случайной величины x . Однако с достаточной для практики точностью при моделировании случайной величины с усеченным нормальным распределением можно обой-

тись без расчетов по формулам. Для определения возможных значений случайной величины с этим распределением можно использовать алгоритм, схема которого приведена на рис. 3.8.

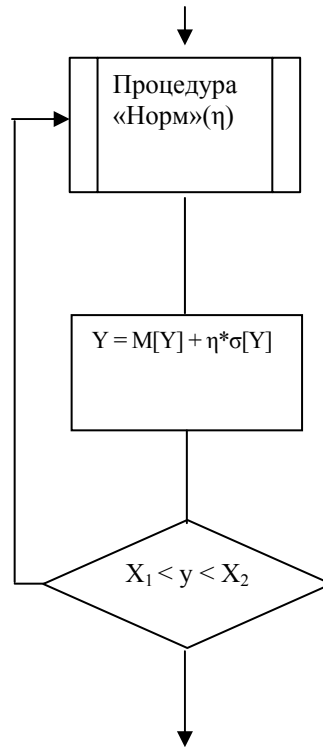


Рис. 3.8. Схема алгоритма моделирования случайной величины с усеченным нормальным распределением

Оператор 1 обращается к процедуре моделирования возможных значений нормированной и центрированной случайной величины η с нормальным распределением. Оператор 2 вычисляет значение случайной величины с заданными параметрами $M(Y)$ и $\sigma(Y)$.

Условный оператор 3 проверяет условие попадания случайной величины y в неусеченную область. При выполнении этого условия значение случайной величины y с усеченным нормальным распределением считается найденным. В противном случае управление в алгоритме передается вновь на вход оператора 1 и генерируется другая случайная величина.

2.2.6. Моделирование случайных величин с произвольным распределением

Пусть случайная величина x задана в интервале (a_0, a_n) кусочно-постоянной функцией $f(x)$. Это значит, что интервал разбит на n частичных интервалов и плотность распределения $f(x)$ на каждом из них постоянна (рис. 3.9).

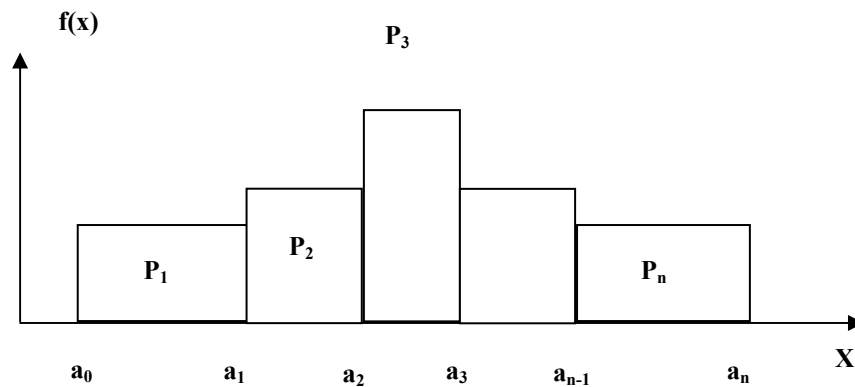


Рис. 3.9. Плотность распределения произвольной функции

Целесообразно выбрать величины a_k так, чтобы вероятности попадания в любой частичный интервал P_k были одинаковы, т.е.

$$\int_{a_{k-1}}^{a_k} f(x)dx = \frac{1}{n}$$

$(k = 1, 2, \dots, n)$.

Из условия постоянства функции на каждом частичном интервале следует, что случайная величина x может быть определена по формуле

$$X = a_{k-1} + z(a_k - a_{k-1}) \quad (k = 1, 2, \dots, n),$$

где z – возможное значение (реализация) случайной величины, равномерно распределенной в интервале $(0,1)$;

a_{k-1} – левая граница частичного интервала;

a_k – правая граница частичного интервала.

Попадание в любой частичный интервал можно рассматривать как событие, входящее в полную группу несовместных событий. Поэтому процедура моделирования в общем случае состоит в следующем:

1. С помощью датчика случайных чисел с равномерным распределением, вырабатывающего величину z , моделируют дискретную случайную величину – номер интервала k .

2. Вторично разыгрывают случайную величину z и определяют возможное значение случайной величины x .

Схема алгоритма показана на рис. 3.10.

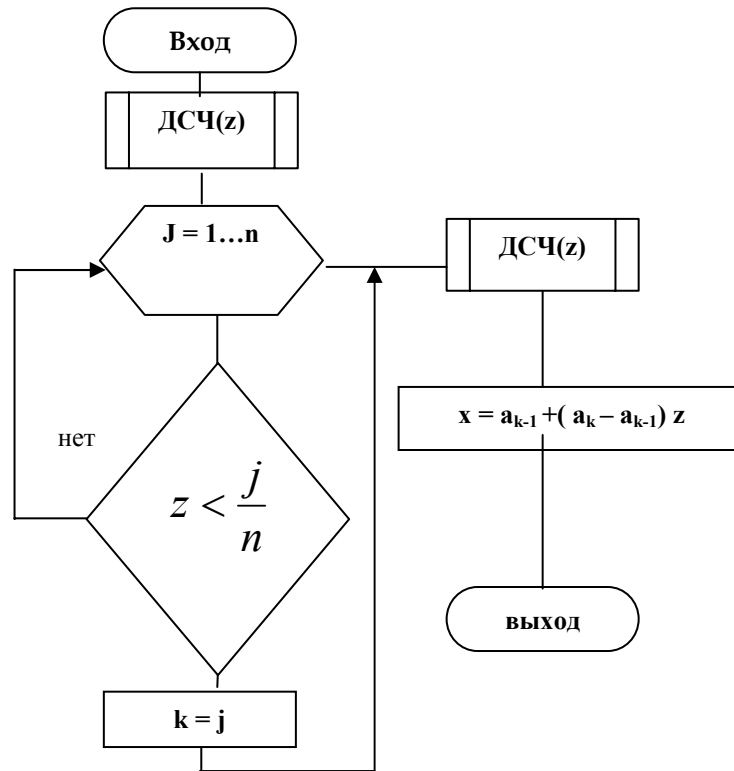


Рис. 3.10. Алгоритм моделирования случайных величин с произвольным распределением

2.2.7. Моделирование случайных величин с заданными параметрами средствами Matlab

При создании имитационной модели средствами Matlab процедура отображения в ней СВ существенно упрощается. Разработчику достаточно иметь представление о том, какие генераторы случайных чисел входят в состав компоненты Toolboxes-Statistics. Как нам уже известно, таких генераторов

более 20. Технология использования в S-модели любого из них одинакова и состоит в выполнении следующих действий:

1. Открыть встроенную справочную систему Matlab (раздел Toolboxes-Statistics).
2. В списке Random Number Generation выбрать функцию, соответствующую требуемому закону распределения.
3. Двойным щелчком ЛКМ на выбранной строке открыть страницу справочника, содержащую описание данного генератора; при этом в верхнем левом поле окна будет выведено название генератора; выделите его с помощью мыши и скопируйте в буфер обмена (используя комбинацию клавиш <Ctrl> + <C>).
4. В блок-диаграмме выбрать блок, в котором будет использоваться генератор, и открыть окно его настроек.
5. Вставить из буфера обмена название генератора (сочетание клавиш <Ctrl> + <V>).
6. Ввести требуемые значения параметров «запуска» генератора.

В качестве примера использования генератора СЧ рассмотрим S-модель, содержащую случайную величину, распределенную по нормальному закону. Согласно теории вероятностей, большинство случайных явлений и процессов, зависящих от многих одновременно действующих факторов, подчиняется именно этому закону.

Пусть имеется вычислительная система, содержащая 2 дисковых накопителя различной емкости: 2,1 Гбайт (назовем его HD1) и 4,3 Гбайт (HD2). Данные поступают на каждый из накопителей от своего источника. Объем очередной «порции» информации является случайной величиной, распределенной по нормальному закону.

Для первого источника закон распределения СВ имеет параметры $m_1=70$ Мб, $v_1=5$ Мб; для второго источника – $m_2 = 120$ Мб, $v_1=10$ Мб. Требуется сравнить эффективность использования накопителей. В качестве показателей эффективности выберем коэффициент использования дискового пространства – $K_{и}$. Эта величина может быть рассчитана как от-

ношение объема памяти, использованного на интервале моделирования, к полной емкости накопителя.

Блок-диаграмма S-модели, позволяющей решить поставленную задачу, показана на рис. 3.11.

Из рисунка видно, что блок-диаграмма состоит из двух достаточно самостоятельных частей. Каждая из них моделирует работу одного из накопителей. Поскольку обе части модели по структуре полностью одинаковы, поясним назначение входящих в них блоков применительно к первой, моделирующей работу накопителя HD1.

- Блок Matlab Fcn играет роль источника данных (или, как будет рассмотрено позднее, – это модель рабочей нагрузки накопителя); в качестве параметра настройки блока указано имя генератора нормального распределения с соответствующими аргументами: *normrnd(70,5)*;
- Блок Discrete - Time Integrator выполняет суммирование объема данных, поступающих от источника в накопитель; все параметры имеют значения, установленные по умолчанию;
- Блок Fcn обеспечивает расчет значения K_i (вычисляемое выражение выводится на пиктограмме блока) для текущего значения модельного времени;
- Блок Display выводит на экран вычисленное значение K_i ; по истечении интервала моделирования оно представляет собой итоговую оценку данного показателя для накопителя HD1.

Немного забегаая вперед (чтобы дать студентам возможность практически опробовать работу модели), отметим, что перед запуском модели целесообразно установить (в меню Simulation) способ изменения модельного времени Fixed-step, а также запретить выдачу предупреждений о неподключенных портах блоков (на вкладке Diagnostics).

По окончании сеанса моделирования в окнах Display будут выведены значения K_i для обоих накопителей практически одновременно.

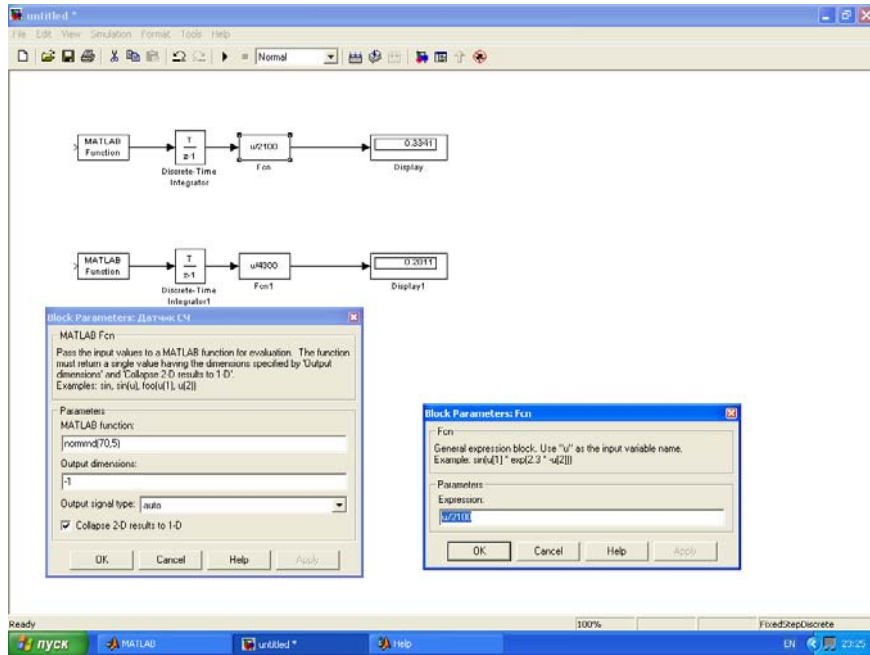


Рис. 3.11. Блок-диаграмма работы накопителей

Раздел II.

Концепция и возможности объектно-ориентированной моделирующей системы

Глава 4. Общие сведения о MATLAB/SIMULINK. Библиотека блоков SIMULINK

Экономическая теория рассматривает в основном динамические проблемы, гипотезы и закономерности.

В математических и технических вузах РФ широко преподается математический программный инструмент Matlab. Для него как надстройки (Toolboxes) разработаны многие спецприложения для анализа различных систем. Он также предоставляет экономистам финансовый пакет FinancialToolbox, связь с Excel – Excellink, связь с Word – Notebook.

Особый интерес для экономистов представляет инструмент Simulink, разработанный специально для моделирования динамических систем. Он имеет библиотеку стандартных графических блоков с встроенными математическими функциями. Иногда его называют инструментом графического (визуального) программирования. Для проведения моделирования достаточно с помощью мышки перетащить из библиотеки блоки в окно модели, соединить их информационными линиями. Создав таким образом модель, запустив ее, можно наблюдать результаты моделирования в окнах графопостроителей и цифровых дисплеев.

При работе с SIMULINK в основном используются файлы трех типов.

M-файлы (с расширением **.m**) – файлы, содержащие тексты программ на языке MATLAB. В виде M-файлов реализованы все библиотечные функции MATLAB. По умолчанию

M-файлы открываются с помощью собственного редактора/отладчика MATLAB.

Mdl-файлы (с расширением **.mdl**) – файлы моделей SIMULINK. Они могут быть открыты либо с помощью SIMULINK (в виде графического окна с блок-схемой), либо в виде текста с помощью редактора/отладчика MATLAB.

MAT-файлы (с расширением **.mat**) – файлы данных, хранящиеся в рабочем пространстве (Workspace) MATLAB. Они либо вводятся вручную в командном окне, либо записываются в эту оперативную память из любого пакета MATLAB или из любого приложения, взаимодействующего с MATLAB, например, Excel или Word. Любые приложения могут читать эти данные.

§1. Запуск MATLAB, интерфейс

Для запуска MATLAB надо дважды щелкнуть левой кнопкой мыши на его ярлыке.

Затем появляется рабочий стол MATLAB с вложенными окнами (рис. 4.1).

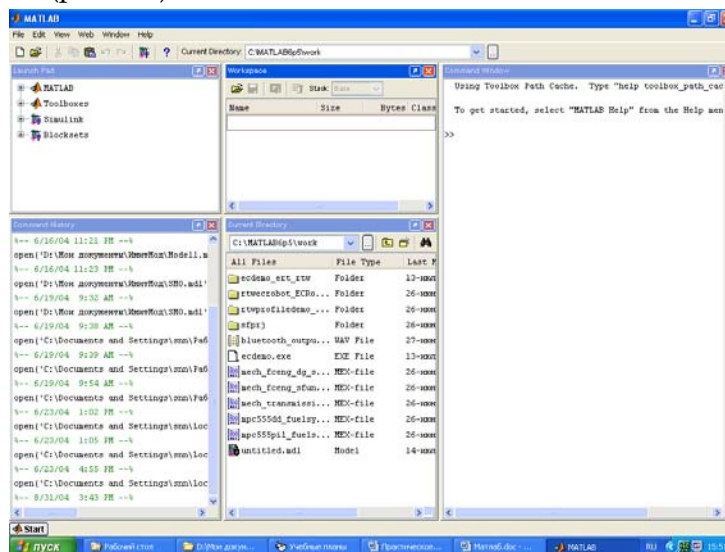


Рис. 4.1. Окна MATLAB

Чтобы не путать рабочие столы MATLAB и Windows, будем называть стол MATLAB главным окном.

Слева в главном окне расположено окно Launch Pad – первое блюдо для освоения MATLAB. Мы видим здесь продукты, которые заказали во время инсталляции. Раскроем папки продуктов и увидим папки Help (текстовая помощь), Demos (демонстрационные примеры), Product Page (Web-страница продуктов фирмы Math/Works Inc. на сайте фирмы в Internet).

В этом же окне можно переключиться на лист Workspace. Это рабочее пространство MATLAB. В этой оперативной памяти сохраняются все данные рабочей сессии MATLAB, Simulink и других инструментов. Их всегда можно просмотреть или обработать любым инструментом.

В центре расположено окно Current Directory – текущий, рабочий справочник (папка). Терминология сохранилась от старой операционной системы MS DOS. Чтобы MATLAB мог увидеть вашу программу (м-файл) или функцию, надо установить текущей папку, в которой находится эта функция.

Внизу слева находится окно Command History (история команд) – это протокол вашей работы.

Справа расположено Command Windows – окно для ввода и исполнения команд.

Окно MATLAB представляет собой стандартное окно Windows-приложения и содержит все основные компоненты такого окна:

- √ строку заголовка с кнопками управления окном;
- √ строку меню (основное меню приложения);
- √ панель инструментов;
- √ рабочее поле;
- √ строку состояния;
- √ вертикальную и горизонтальную полосы прокрутки.

Строка меню MATLAB содержит следующие команды:

File (файл) – команды для работы с файлами и настройки системы;

Edit (правка) – команды редактирования информации, отображенной в рабочем поле окна;

View (вид) – команды управления форматом окна;

Web – связь по интернету с фирмой по многим вопросам приобретения, регистрации, консультаций, применения MATLAB;

Windows (окно) – список открытых окон приложения;

Help (справка) – команды вызова средств поддержки пользователя.

Команды меню File.

New – создать;

Open – открыть;

Close Launch Pad – закрыть окно начального знакомства MATLAB;

Import Data – прием данных из других приложений;

Save Workspace As – сохранить рабочую область как...;

Set Path – выбор рабочей папки;

Preferences (предпочтения) – настройка форматов чисел, экрана и других параметров для умолчания, что в офисных продуктах обычно делается в меню Сервис.

Print – печать;

Print Selection – печать выделенного.

Далее список файлов, открывавшихся в прошлом и текущем сеансе работы с MATLAB. Вначале этот список пуст.

Exit MATLAB – выход из MATLAB.

Команды меню Edit.

Они аналогичны командам офисных пакетов.

Команды Clear стирают содержимое окон: командного, истории и рабочего пространства.

Команды меню View (вид).

Здесь расположены команды показа или скрытия окон, а также команда Undock, которая позволяет вывести любое текущее окно из главного окна на рабочий стол Windows.

Help – справочная система.

Help browser – это Web-browser, интегрированный в экран MATLAB и отображающий документы в формате интернета HTML. Все продукты Math\Works Inc. можно получить по интернету.

Предоставляется также справочная система в формате PDF. Справочник по функциям MATLAB содержится в разделе Reference. В нем сведения о назначении и параметрах, а также примеры использования функций MATLAB, входящих в состав рабочей конфигурации пакета.

Изменение конфигурации приводит к изменению списка функций, по которым может быть получена справка.

Чтобы получить полную информацию по интересующему разделу, достаточно щелкнуть в соответствующей строке. В поле просмотра появится список функций, входящих в этот раздел, с указанием назначения каждой из них. Щелкнув на нужной функции, можно получить о ней более подробные сведения.

Ниже строки меню расположены кнопки команд меню. Они стандартны для Windows-приложений. Но предпоследняя (перед кнопкой «?») зеленая с красным левым нижним углом – это кнопка вызова Simulink.

§2. Editor/debugger – редактор/отладчик программ

Для автоматизации управления экономическими экспериментами с моделями Simulink приходится писать программы на языке MATLAB. Программы (м-файлы) пишутся и отлаживаются в редакторе/отладчике. Он вызывается, когда из меню Файл MATLAB мы открываем новый или существующий м-файл (рис. 4.2).

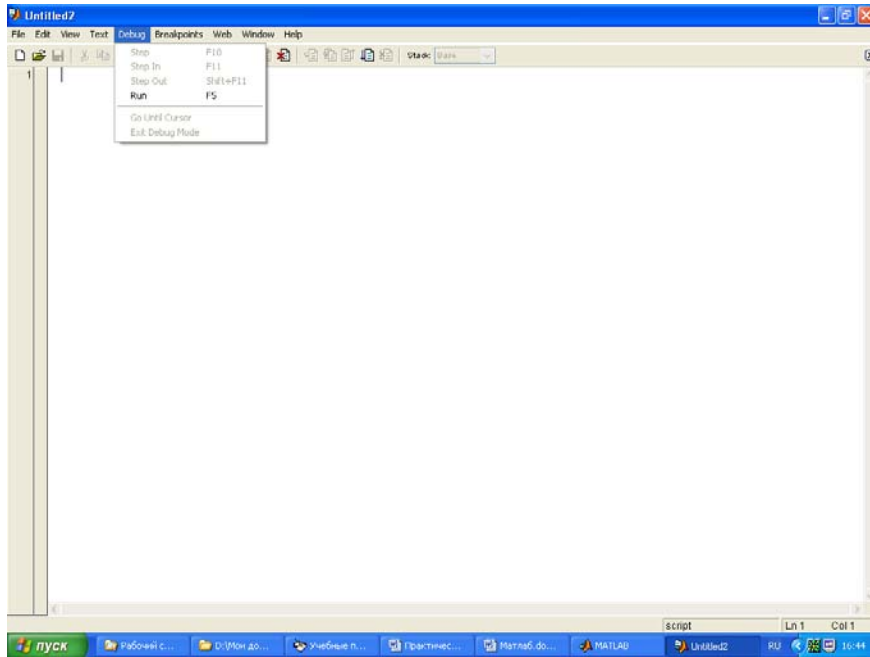


Рис. 4.2. Окно для написания m-файла

Программа пишется как в обычном текстовом редакторе. В меню имеются лишь два раздела, относящиеся к отладке: Debug и Breakpoints.

Для выполнения лабораторных работ на отлаженной модели экономисту наиболее полезна команда Run из раздела Debug. Она запускает программу на исполнение и манипуляции с моделью Simulink. Этой же программой обрабатываются и отображаются результаты экспериментов.

Breakpoints – точки остановки. Ими помечают строки программы для остановки и анализа поэтапного исполнения программы при ее отладке.

Более подробно рассматривать работу с редактором/отладчиком будем на последующих занятиях.

§3. Простые вычисления в командном режиме

В MATLAB можно различать два режима работы: вычисления в командном режиме и исполнение программ, записанных на его языке.

В командном окне представлен символ «>>», означающий готовность MATLAB к исполнению команд оператора. Они выполняются как в любом калькуляторе, например, Бэйсика или Excel.

В командном окне (рис. 4.3) мы ввели выражение $2*2$. Чтобы его вычислить (исполнить), нажмите клавишу Enter. Получим ответ `ans = 4` (от англ. answer – ответ). Затем MATLAB показал знак готовности к исполнению новых команд «>>».

Вместо числа мы можем ввести матрицу или вектор, например:

$$u = [1,2,3,4].$$

Нажмем клавишу исполнения Enter и получим ответ

$$u = 1 \ 2 \ 3 \ 4.$$

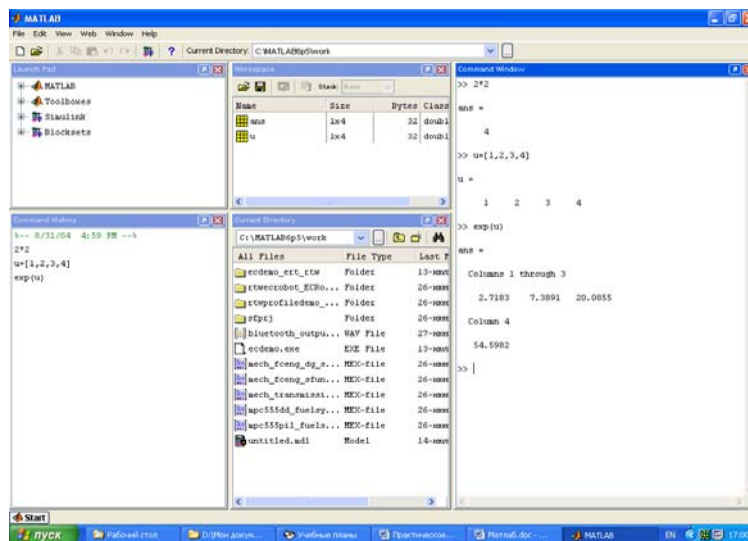


Рис. 4.3. Простые вычисления в командном режиме

MATLAB может вычислять практически все математические функции. Например, на нашем рисунке в окне он использует наш вектор \mathbf{u} и вычисляет вектор экспоненциальных функций $\exp(\mathbf{u})$ в векторной переменной `ans`.

Обратите внимание на информацию, выводимую в окнах `Workspace`, `Command History`.

Можно ввести последовательность команд. Если команда не заканчивается символом точки с запятой (`;`), то она выполняется сразу же после нажатия клавиши `Enter`.

Использование разделителя в виде точки с запятой позволяет вводить в рабочем поле последовательность команд, которая будет выполнена только в том случае, если после очередной команды не стоит этот символ. Если выполнение команды приводит к вычислению некоторого значения (скалярного или матрицы), то оно запоминается в рабочей области MATLAB в переменной с именем `ans`. Значение, занесенное в переменную `ans`, выводится на экран сразу после вычисления в форме `ans = значение (число, вектор, матрица)`.

§4. Введение в Simulink

Программа Simulink является приложением к пакету MATLAB. При моделировании с использованием Simulink реализуется принцип визуального программирования, в соответствии с которым пользователь на экране из библиотеки стандартных блоков создает модель устройства, процесса или системы и осуществляет расчеты. При этом, в отличие от классических способов моделирования, пользователю не нужно досконально изучать язык программирования и численные методы математики, а достаточно общих знаний, требующихся при работе на компьютере, и, естественно, знаний той предметной области, в которой он работает.

Simulink является достаточно самостоятельным инструментом MATLAB и при работе с ним совсем не требуется знать сам MATLAB и остальные его приложения. С другой стороны, доступ к функциям MATLAB и другим его инструментам остается открытым и их можно использовать в

Simulink. При работе с Simulink пользователь имеет возможность модернизировать библиотечные блоки, создавать свои собственные, а также составлять новые библиотеки блоков.

При моделировании пользователь может выбирать метод решения дифференциальных уравнений, а также способ изменения модельного времени (с фиксированным или переменным шагом). В ходе моделирования имеется возможность следить за процессами, происходящими в системе. Для этого используются специальные устройства наблюдения, входящие в состав библиотеки Simulink. Результаты моделирования могут быть представлены в виде графиков или таблиц.

Преимущество Simulink заключается также в том, что она позволяет пополнять библиотеки блоков с помощью подпрограмм, написанных как на языке MATLAB, так и на языках C++, Fortran и Ada.

§5. Работа с Simulink

Для запуска программы необходимо предварительно запустить пакет MATLAB. Основное окно пакета MATLAB показано на рис. 4.4. Там же показана подсказка, появляющаяся в окне при наведении указателя мыши на ярлык Simulink в панели инструментов.

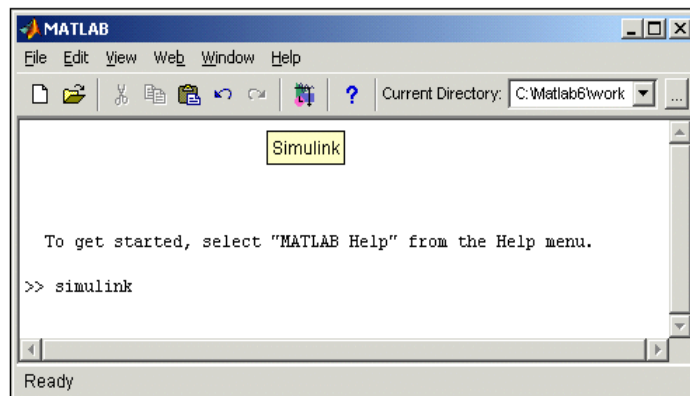


Рис. 4.4. Основное окно программы MATLAB

После открытия основного окна программы MATLAB нужно запустить программу Simulink. Это можно сделать одним из трех способов:

- Нажать кнопку (Simulink) на панели инструментов командного окна MATLAB.
- В командной строке главного окна MATLAB напечатать Simulink и нажать клавишу Enter на клавиатуре.
- Выполнить команду Open... в меню File и открыть файл модели (mdl-файл).

Последний вариант удобно использовать для запуска уже готовой и отлаженной модели, когда требуется лишь провести расчеты и не нужно добавлять новые блоки в модель. Использование первого и второго способов приводит к открытию окна обозревателя разделов библиотеки Simulink (рис. 4.5).

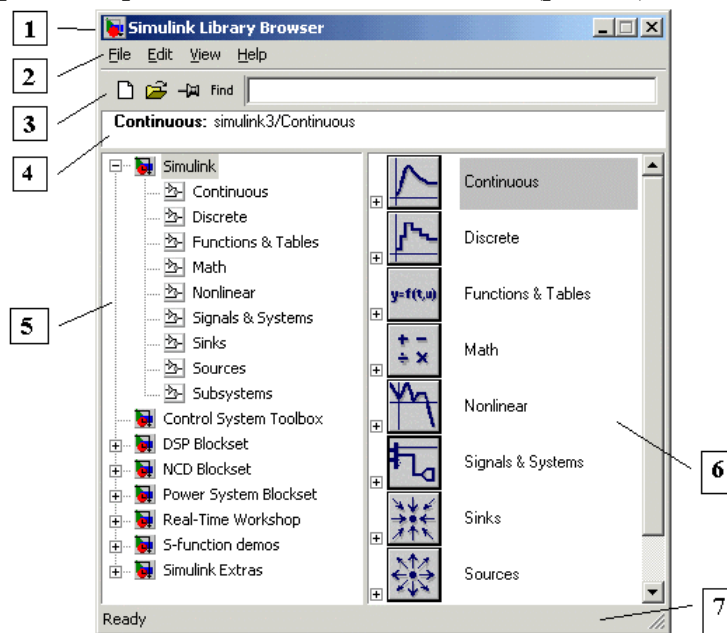


Рис. 4.5. Окно обозревателя разделов библиотеки Simulink

§6. Обзорщик разделов библиотеки Simulink

Окно обзорщика библиотеки блоков содержит следующие элементы:

1. Заголовок с названием окна – Simulink Library Browser.
2. Меню с командами File, Edit, View, Help.
3. Панель инструментов с ярлыками наиболее часто используемых команд.
4. Окно комментария для вывода поясняющего сообщения о выбранном блоке.
5. Список разделов библиотеки, реализованный в виде дерева.
6. Окно содержимого раздела библиотеки (список вложенных разделов библиотеки или блоков).
7. Строка состояния, содержащая подсказку по выполняемому действию.

Библиотека Simulink содержит следующие основные разделы:

1. Continuous – линейные блоки.
2. Discrete – дискретные блоки.
3. Functions & Tables – функции и таблицы.
4. Math – блоки математических операций.
5. Nonlinear – нелинейные блоки.
6. Signals & Systems – сигналы и системы.
7. Sinks – регистрирующие устройства.
8. Sources – источники сигналов и воздействий.
9. Subsystems – блоки подсистем.

Список разделов библиотеки Simulink представлен в виде дерева, и правила работы с ним являются общими для списков такого вида:

- Пиктограмма свернутого узла дерева содержит символ «+», а пиктограмма развернутого содержит символ «-».
- Для того чтобы развернуть или свернуть узел дерева, достаточно щелкнуть на его пиктограмме левой клавишей мыши (ЛКМ).

При выборе соответствующего раздела библиотеки в правой части окна отображается его содержимое (рис. 4.6).

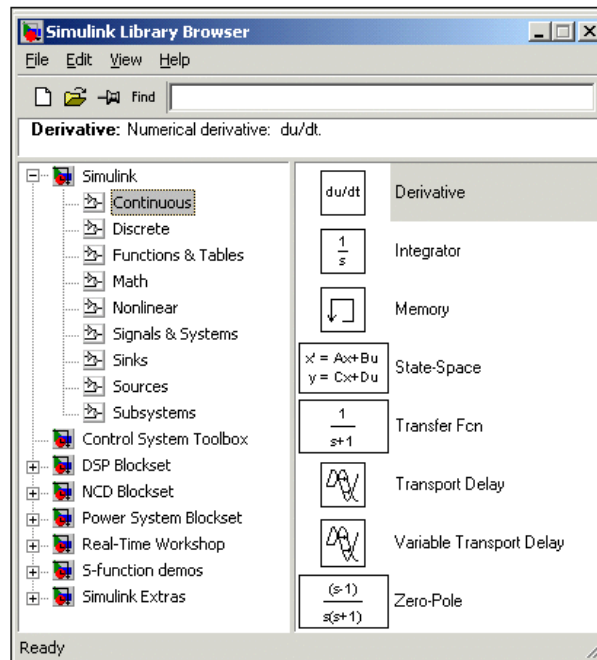


Рис. 4.6. Окно обозревателя с набором блоков раздела библиотеки

Для работы с окном используются команды, собранные в меню. Меню обозревателя библиотек содержит следующие пункты:

- File (Файл) – работа с файлами библиотек.
- Edit (Редактирование) – добавление блоков и их поиск (по названию).
- View (Вид) – управление показом элементов интерфейса.
- Help (Справка) – вывод окна справки по обозревателю библиотек.

Для работы с обозревателем можно также использовать кнопки на панели инструментов.

§7. Создание модели

Для создания модели в среде SIMULINK необходимо последовательно выполнить ряд действий:

1. Создать новый файл модели с помощью команды File/New/Model или используя кнопку на панели инструментов (здесь и далее, с помощью символа «/», указаны пункты меню программы, которые необходимо последовательно выбрать для выполнения указанного действия). Вновь созданное окно модели показано на рис. 4.7.

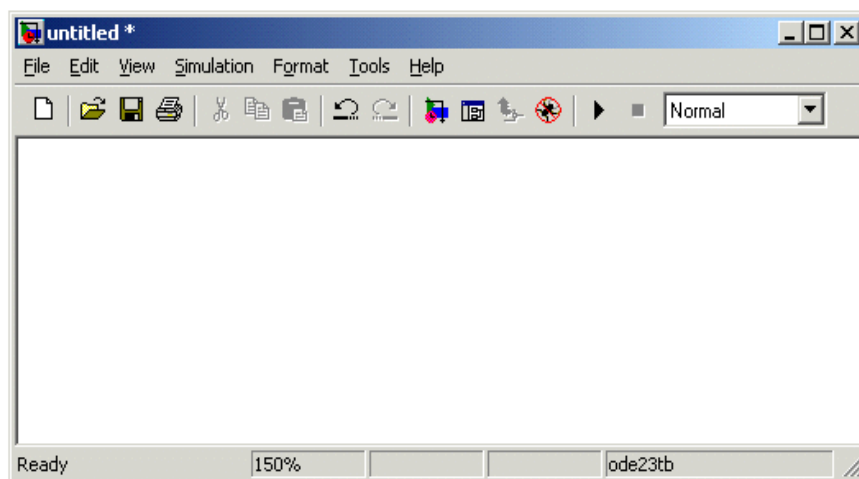


Рис. 4.7. Пустое окно модели

2. Расположить блоки в окне модели.

Для этого необходимо открыть соответствующий раздел библиотеки (например, Sources – Источники). Далее, указав курсором на требуемый блок и нажав на левую клавишу мыши, – «перетащить» блок в созданное окно. **Клавишу мыши нужно держать нажатой.** На рис. 4.8 показано окно модели, содержащее блоки.

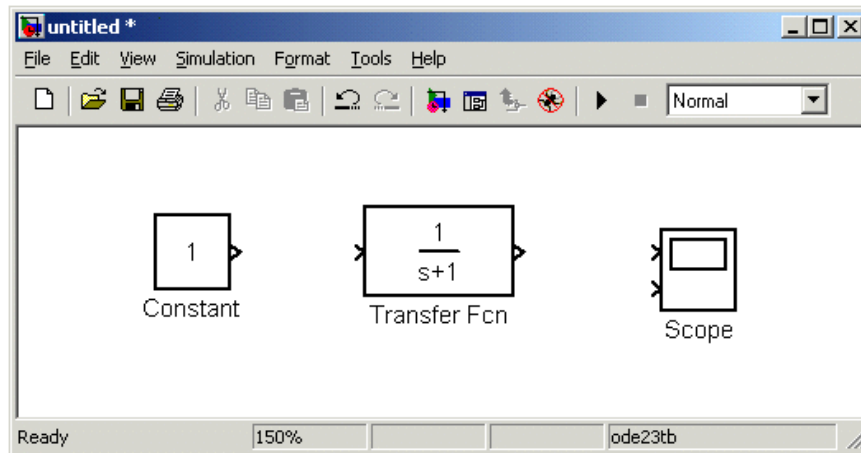


Рис. 4.8. Окно модели, содержащее блоки

Для удаления блока необходимо выбрать блок (указать курсором на его изображение и нажать левую клавишу мыши), а затем нажать клавишу Delete на клавиатуре.

Для изменения размеров блока требуется выбрать блок, установить курсор в один из углов блока и, нажав левую клавишу мыши, изменить размер блока (курсор при этом превратится в двухстороннюю стрелку).

3. Далее, если это требуется, нужно изменить параметры блока, установленные программой «по умолчанию».

Для этого необходимо дважды щелкнуть левой клавишей мыши, указав курсором на изображение блока. Откроется окно редактирования параметров данного блока. При задании численных параметров следует иметь в виду, что в качестве десятичного разделителя должна использоваться точка, а не запятая. После внесения изменений нужно закрыть окно кнопкой ОК. На рис. 4.9 в качестве примера показаны блок, моделирующий передаточную функцию, и окно редактирования параметров данного блока.

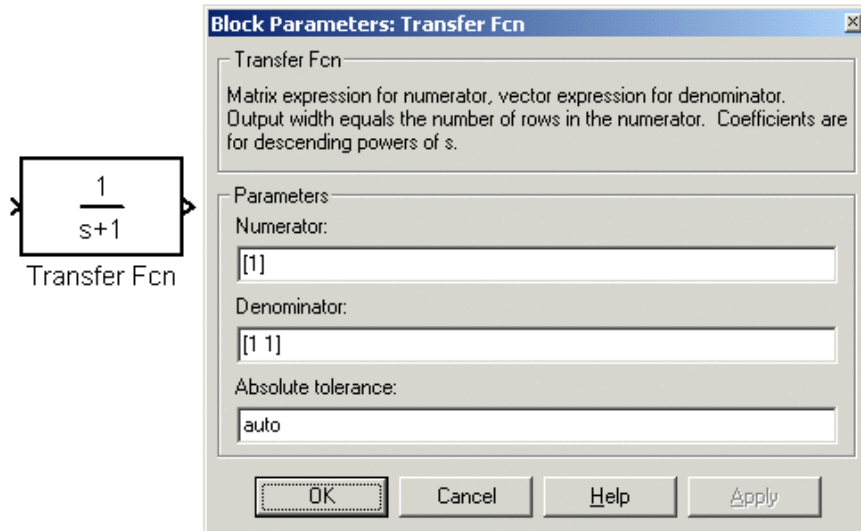


Рис. 4.9. Блок, моделирующий передаточную функцию, и окно редактирования параметров блока

4. После установки на схеме всех блоков из требуемых библиотек нужно выполнить соединение элементов схемы (рис. 4.10). Для соединения блоков необходимо указать курсором на «выход» блока, а затем нажать и, не отпуская левую клавишу мыши, провести линию к входу другого блока. После чего отпустить клавишу. В случае правильного соединения изображение стрелки на входе блока изменяет цвет. Для создания точки разветвления в соединительной линии нужно подвести курсор к предполагаемому узлу и, нажав *правую* клавишу мыши, протянуть линию. Для удаления линии требуется выбрать линию (так же, как это выполняется для блока), а затем нажать клавишу Delete на клавиатуре.

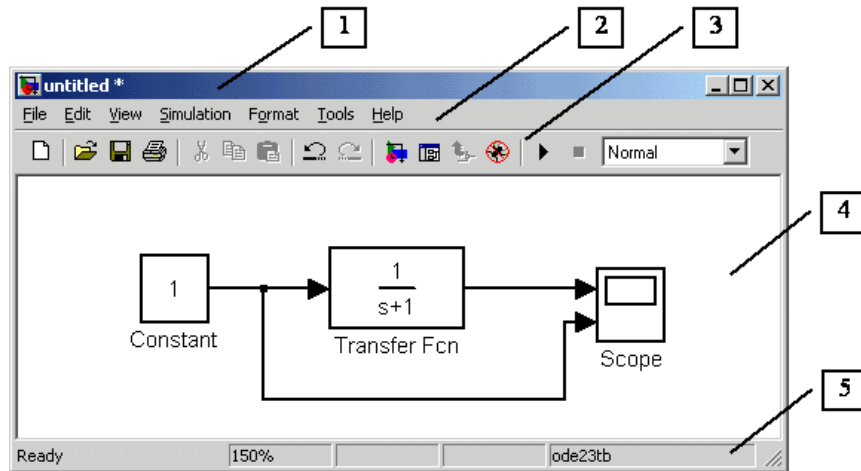


Рис. 4.10. Схема модели с выполнением соединением блоков

5. После составления расчетной схемы необходимо сохранить ее в виде файла на диске, выбрав пункт меню File/Save As... в окне схемы и указав папку и имя файла. Следует иметь в виду, что имя файла не должно превышать 32 символов, должно начинаться с буквы и не может содержать символы кириллицы и спецсимволы. Это же требование относится и к пути файла (к тем папкам, в которых сохраняется файл). При последующем редактировании схемы можно пользоваться пунктом меню File/Save. При повторных запусках программы SIMULINK загрузка схемы осуществляется с помощью меню File/Open... в окне обозревателя библиотеки или из основного окна MATLAB.

§8. Окно модели

Окно модели содержит следующие элементы (см. рис. 4.10):

1. Заголовок с названием окна. Вновь созданному окну присваивается имя Untitled с соответствующим номером.
2. Меню с командами File, Edit, View и т.д.

3. Панель инструментов.
4. Окно для создания схемы модели.
5. Строка состояния, содержащая информацию о текущем состоянии модели.

Меню окна содержит команды для редактирования модели, ее настройки и управления процессом расчета, работы файлами и т.п.:

- File (Файл) – работа с файлами моделей.
- Edit (Редактирование) – изменение модели и поиск блоков.
- View (Вид) – управление показом элементов интерфейса.
- Simulation (Моделирование) – задание настроек для моделирования и управление процессом расчета.
- Format (Форматирование) – изменение внешнего вида блоков и модели в целом.
- Tools (Инструментальные средства) – применение специальных средств для работы с моделью (отладчик, линейный анализ и т.п.).
- Help (Справка) – вывод окон справочной системы.

Для работы с моделью можно также использовать кнопки на панели инструментов (рис. 4.11).

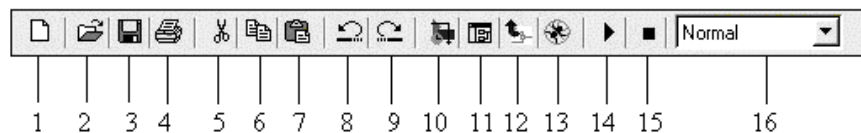


Рис. 4.11. Панель инструментов окна модели

Кнопки панели инструментов имеют следующее назначение:

1. New Model – открыть новое (пустое) окно модели.
2. Open Model – открыть существующий mdl-файл.
3. Save Model – сохранить mdl-файл на диске.
4. Print Model – вывод на печать блок-диаграммы модели.
5. Cut – вырезать выделенную часть модели в буфер промежуточного хранения.

6. Copy – скопировать выделенную часть модели в буфер промежуточного хранения.
7. Paste – вставить в окно модели содержимое буфера промежуточного хранения.
8. Undo – отменить предыдущую операцию редактирования.
9. Redo – восстановить результат отмененной операции редактирования.
10. Library Browser – открыть окно обозревателя библиотек.
11. Toggle Model Browser – открыть окно обозревателя модели.
12. Go to parent system – переход из подсистемы в систему высшего уровня иерархии («родительскую систему»). Команда доступна только, если открыта подсистема.
13. Debug – запуск отладчика модели.
14. Start/Pause/Continue Simulation – запуск модели на исполнение (команда Start).
15. Stop – закончить моделирование. Кнопка становится доступной после начала моделирования, а также после выполнения команды Pause.
16. Normal/Accelerator – обычный/ускоренный режим расчета. Инструмент доступен, если установлено приложение Simulink Performance Tool.

В нижней части окна модели находится строка состояния, в которой отображаются краткие комментарии к кнопкам панели инструментов, а также к пунктам меню, когда указатель мыши находится над соответствующим элементом интерфейса. Это же текстовое поле используется и для индикации состояния Simulink: Ready (Готов) или Running (Выполнение). В строке состояния отображаются также:

- масштаб отображения блок-диаграммы (в процентах, исходное значение равно 100%);
- индикатор степени завершенности сеанса моделирования (появляется после запуска модели);
- текущее значения модельного времени (выводится также только после запуска модели);
- используемый алгоритм расчета состояний модели (метод решения).

§9. Основные приемы подготовки и редактирования модели

Добавление текстовых надписей

Для повышения наглядности модели удобно использовать текстовые надписи. Для создания надписи нужно указать мышью место надписи и дважды щелкнуть левой клавишей мыши. После этого появится прямоугольная рамка с курсором ввода. Аналогичным образом можно изменить и подписи к блокам моделей. На рис. 4.12 показаны текстовая надпись и изменение надписи в блоке передаточной функции. Следует иметь в виду, что различные версии программы (Simulink 4) по-разному адаптированы к использованию кириллических шрифтов и применение их может иметь самые разные последствия: отображение надписей в нечитаемом виде, обрезание надписей, сообщения об ошибках, а также невозможность открыть модель после ее сохранения. Поэтому применение надписей на русском языке для Simulink не всегда желательно.

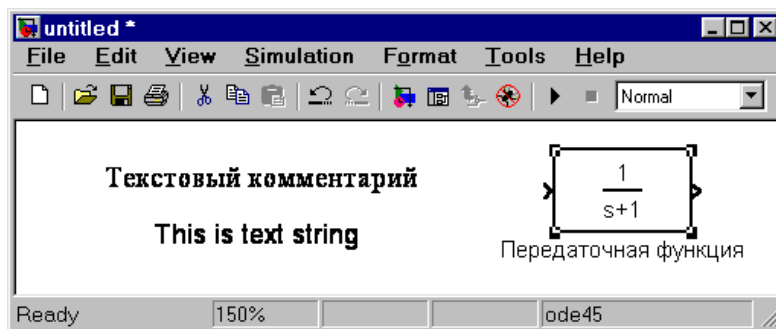


Рис. 4.12. Текстовая надпись и изменение надписи
в Transfer Function

Выделение объектов

Для выполнения какого-либо действия с элементом модели (блоком, соединительной линией, надписью), этот элемент необходимо сначала выделить.

Выделение объектов проще всего осуществляется мышью. Для этого необходимо установить курсор мыши на нужном объекте и щелкнуть левой клавишей мыши. Произойдет выделение объекта. Об этом будут свидетельствовать маркеры по углам объекта. Можно также выделить несколько объектов. Для этого надо установить курсор мыши вблизи группы объектов, нажать левую клавишу мыши и, не отпуская ее, начать перемещать мышью. Появится пунктирная рамка, размеры которой будут изменяться при перемещении мыши. Все охваченные рамкой объекты становятся выделенными. Выделить все объекты также можно используя команду Edit/Select All. После выделения объекта его можно копировать или перемещать в буфер промежуточного хранения, извлекать из буфера, а также удалять, используя стандартные приемы работы в Windows-программах.

Копирование и перемещение объектов в буфер промежуточного хранения

Для копирования объекта в буфер его необходимо предварительно выделить, а затем выполнить команду Edit/Сору.

Для вырезания объекта в буфер его необходимо предварительно выделить, а затем выполнить команду Edit/Cut или воспользоваться инструментом на панели инструментов. При выполнении данных операций следует иметь в виду, что объекты помещаются в собственный буфер MATLAB и недоступны из других приложений. Использование команды Edit/Сору model to Clipboard позволяет поместить *графическое изображение* модели в буфер Windows и, соответственно, делает его доступным для остальных программ.

Копирование можно выполнить и таким образом: нажать *правую* клавишу мыши и, не отпуская ее, переместить объект. При этом будет создана копия объекта, которую можно переместить в необходимое место.

Вставка объектов из буфера промежуточного хранения

Для вставки объекта из буфера необходимо предварительно указать место вставки, щелкнув левой клавишей мыши в предполагаемом месте вставки, а затем выполнить команду Edit/Paste или воспользоваться инструментом на панели инструментов.

Удаление объектов

Для удаления объекта его необходимо предварительно выделить, а затем выполнить команду Edit/Clear или воспользоваться клавишей Delete на клавиатуре. Следует учесть, что команда Clear удаляет блок без помещения его в буфер обмена. Однако эту операцию можно отменить командой меню File/Undo.

Соединение блоков

Для соединения блоков необходимо сначала установить курсор мыши на выходной порт одного из блоков. Курсор при этом превратится в большой крест из тонких линий (рис. 4.13). Держа нажатой левую кнопку мыши, нужно переместить курсор ко входному порту нужного блока. Курсор мыши примет вид креста из тонких сдвоенных линий (рис. 4.14). После создания линии необходимо отпустить левую клавишу мыши. Свидетельством того, что соединение создано, будет жирная стрелка у входного порта блока. Выделение линии производится точно так же, как и выделение блока, – одинарным щелчком левой клавиши мыши. Черные маркеры, расположенные в узлах соединительной линии, будут говорить о том, что линия выделена.

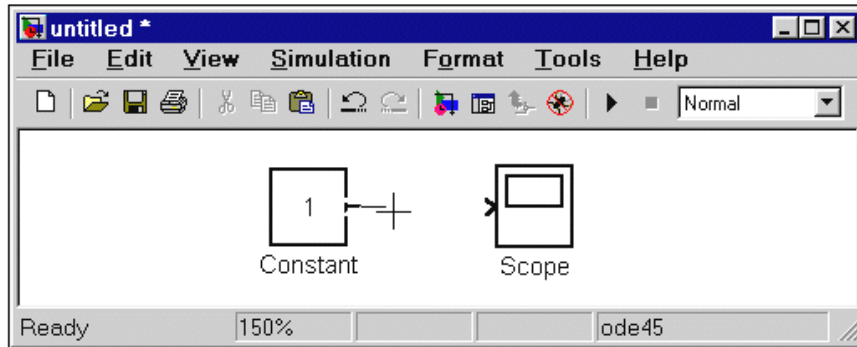


Рис. 4.13. Начало создания соединения

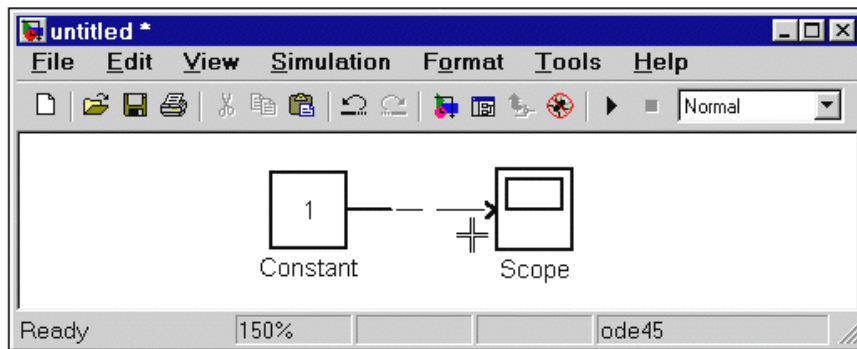


Рис. 4.14. Завершение создания соединения

Создание петли линии соединения выполняется так же, как перемещение блока. Линия соединения выделяется, и затем нужная часть линии перемещается (рис. 4.15).

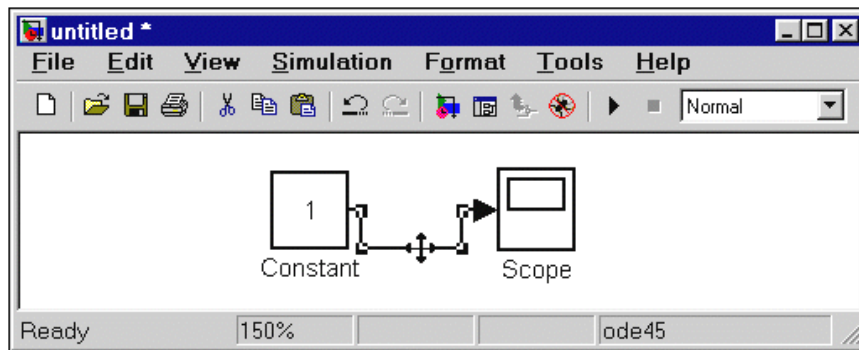


Рис. 4.15. Создание петли в соединительной линии

Удаление соединений выполняется так же, как и удаление любых других объектов.

Изменение размеров блоков

Для изменения размера блока он выделяется, после чего курсор мыши надо установить на один из маркеров по углам блока. После превращения курсора в двухстороннюю стрелку, необходимо нажать левую клавишу мыши и растянуть (или сжать) изображения блока (рис. 4.16). Размеры надписей блока при этом не изменяются.

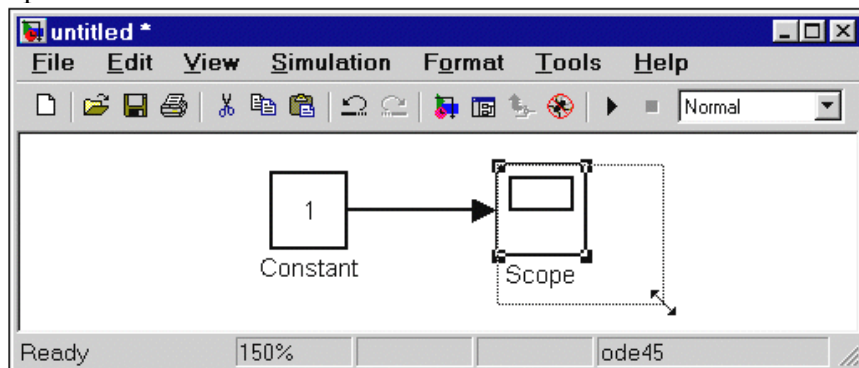


Рис. 4.16. Изменение размера блока

Перемещение блоков

Любой блок модели можно переместить, выделив его и передвинув, держа нажатой левую клавишу мыши. Если ко входам и выходам блока подведены соединительные линии, то они не разрываются, а лишь сокращаются или увеличиваются в длину. В соединении можно также вставить блок, имеющий один вход и один выход. Для этого его нужно расположить в требуемом месте соединительной линии.

Использование команд Undo и Redo

В процессе освоения программы пользователь может совершать действия, кажущиеся ему необратимыми (например, случайное удаление части модели, копирование и т.д.). В этом случае следует воспользоваться командой Undo – отмена последней операции. Команду можно вызвать с помощью кнопки в панели инструментов окна модели или из меню Edit. Для восстановления отмененной операции используется команда Redo.

Форматирования объектов

В меню Format (так же, как и в контекстном меню, вызываемом нажатием правой клавиши мыши на объекте) находится набор команд форматирования блоков. Команды форматирования разделяются на несколько групп.

1. Изменение отображения надписей:

- Font – форматирование шрифта надписей и текстовых блоков.
- Text alignment – выравнивание текста в текстовых надписях.
- Flip name – перемещение подписи блока.
- Show/Hide name – отображение или скрытие подписи блока.

2. Изменение цветов отображения блоков:

- Foreground color – выбор цвета линий для выделенных блоков.
- Background color – выбор цвета фона выделенных блоков.
- Screen color – выбор цвета фона для всего окна модели.

3. Изменение положения блока и его вида:
- Flip block – зеркальное отображение относительно вертикальной оси симметрии.
 - Rotate block – поворот блока на 90° по часовой стрелке.
 - Show drop shadow – показ тени от блока.
 - Show port labels – показ меток портов.
4. Прочие установки:
- Library link display – показ связей с библиотеками.
 - Sample time colors – выбор цвета блока индикации времени.
 - Wide nonscalar lines – увеличение/уменьшение ширины не скалярных линий.
 - Signal dimensions – показ размерности сигналов.
 - Port data types – показ данных о типе портов.
 - Storage class – класс памяти. Параметр, устанавливаемый при работе Real-Time Workshop.

§10. Библиотека блоков SIMULINK

10.1. Sources – источники сигналов

Источник постоянного сигнала Constant

Назначение:

Задаёт постоянный по уровню сигнал.

Параметры:

1. Constant value – постоянная величина.
2. Interpret vector parameters as 1-D – интерпретировать вектор параметров как одномерный (при установленном флажке). Данный параметр встречается у большинства блоков библиотеки Simulink. В дальнейшем он рассматриваться не будет.

Значение константы может быть действительным или комплексным числом, вычисляемым выражением, вектором или матрицей.

Рис. 4.17 иллюстрирует применение этого источника и измерение его выходного сигнала с помощью цифрового индикатора Display.

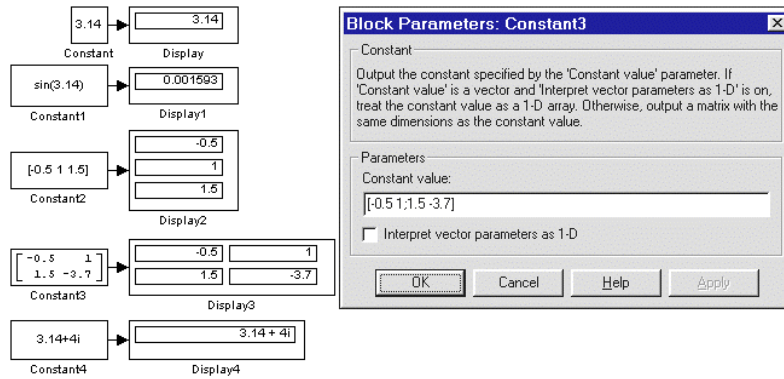


Рис. 4.17. Источник постоянного воздействия Constant

10.2. Sinks – приемники сигналов

10.2.1. Осциллограф Scope

Назначение:

Строит графики исследуемых сигналов в функции времени. Позволяет наблюдать за изменениями сигналов в процессе моделирования.

Изображение блока и окно для просмотра графиков показаны на рис. 4.18.

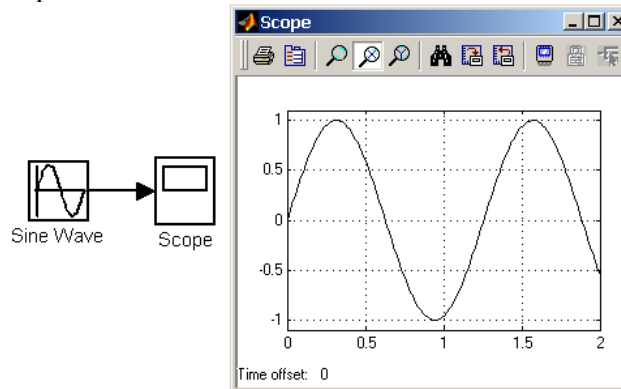


Рис. 4.18. Осциллограф Scope

Для того чтобы открыть окно просмотра сигналов, необходимо выполнить двойной щелчок левой клавишей мыши на изображении блока. Это можно сделать на любом этапе расчета (как до начала расчета, так и после него, а также во время расчета). В том случае, если на вход блока поступает векторный сигнал, то кривая для каждого элемента вектора строится отдельным цветом.

Настройка окна осциллографа выполняется с помощью панелей инструментов (рис. 4.19).

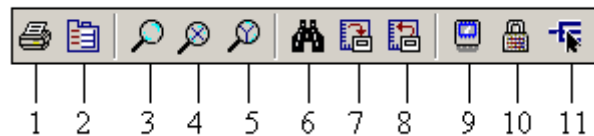








Рис. 4.19. Панель инструментов блока Scope

Панель инструментов содержит 11 кнопок:

1. Print – печать содержимого окна осциллографа.
2. Parameters – доступ к окну настройки параметров.
3. Zoom – увеличение масштаба по обеим осям.
4. Zoom X-axis – увеличение масштаба по горизонтальной оси.
5. Zoom Y-axis – увеличение масштаба по вертикальной оси.
6. Autoscale – автоматическая установка масштабов по обеим осям.
7. Save current axes settings – сохранение текущих настроек окна.
8. Restore saved axes settings – установка ранее сохраненных настроек окна.
9. Floating scope – перевод осциллографа в «свободный» режим.
10. Lock/Unlock axes selection – закрепить/разорвать связь между текущей координатной системой окна и отображаемым сигналом. Инструмент доступен, если включен режим Floating scope.
11. Signal selection – выбор сигналов для отображения. Инструмент доступен, если включен режим Floating scope.

Изменение масштабов отображаемых графиков можно выполнять несколькими способами:

1. Нажать соответствующую кнопку (,  или ) и щелкнуть один раз левой клавишей мыши в нужном месте графика. Произойдет 2,5-кратное увеличение масштаба.

2. Нажать соответствующую кнопку (,  или ) и, нажав левую клавишу мыши, с помощью динамической рамки или отрезка указать область графика для увеличенного изображения (рис. 4.20).

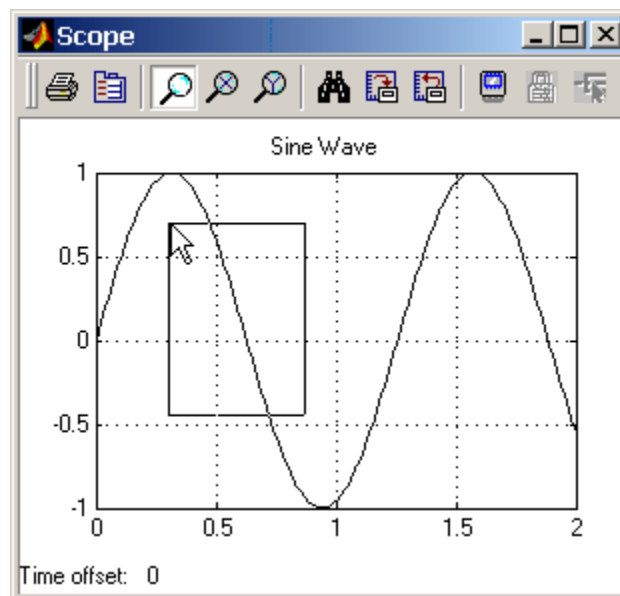


Рис. 4.20. Увеличение масштаба графика

3. Щелкнуть правой клавишей мыши в окне графиков и, выбрать команду *Axis properties...* в контекстном меню. Откроется окно свойств графика, в котором с помощью параметров *Y-min* и *Y-max* можно указать предельные значения вертикальной оси. В этом же окне можно указать заголовок графика

фика (Title), заменив выражение %<SignalLabel> в строке ввода. Окно свойств показано на рис. 4.21.

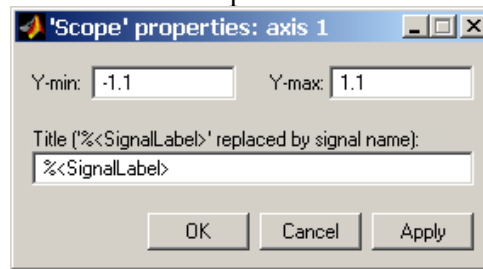



Рис. 4.21. Окно свойств графика

Параметры:

Параметры блока устанавливаются в окне 'Scope' parameters, которое открывается с помощью инструмента  (Parameters) панели инструментов. Окно параметров имеет две вкладки:

General – общие параметры.

Data history – параметры сохранения сигналов в рабочей области MATLAB.

Вкладка общих параметров показана на рис. 4.22.

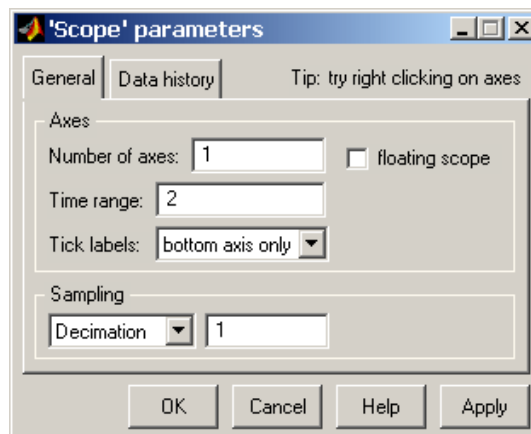


Рис. 4.22. Вкладка общих параметров General

На вкладке General задаются следующие параметры:

1. Number of axes – число входов (систем координат) осциллографа. При изменении этого параметра на изображении блока появляются дополнительные входные порты.

2. Time range – величина временного интервала, для которого отображаются графики. Если время расчета модели превышает заданное параметром Time range, то вывод графика производится порциями, при этом интервал отображения каждой порции графика равен заданному значению Time range.

3. Tick labels – вывод/скрытие осей и меток осей. Может принимать три значения (выбираются из списка):

- all – подписи для всех осей,
- none – отсутствие всех осей и подписей к ним,
- bottom axis only – подписи горизонтальной оси только для нижнего графика.

4. Sampling – установка параметров вывода графиков в окне. Задаёт режим вывода расчетных точек на экран. При выборе Decimation кратность вывода устанавливается числом, задающим шаг выводимых расчетных точек.

5. Floating scope – перевод осциллографа в «свободный» режим (при установленном флажке).

На вкладке Data history (рис. 4.23) задаются следующие параметры:

1. Limit data points to last – максимальное количество отображаемых расчетных точек графика. При превышении этого числа начальная часть графика обрезается. В том случае, если флажок параметра Limit data points to last не установлен, то Simulink автоматически увеличит значение этого параметра для отображения всех расчетных точек.

2. Save data to workspace – сохранение значений сигналов в рабочей области MATLAB.

3. Variable name – имя переменной для сохранения сигналов в рабочей области MATLAB.

4. Format – формат данных при сохранении в рабочей области MATLAB. Может принимать значения:

- Array – массив,
- Structure – структура,
- Structure with time – структура с дополнительным полем «время».

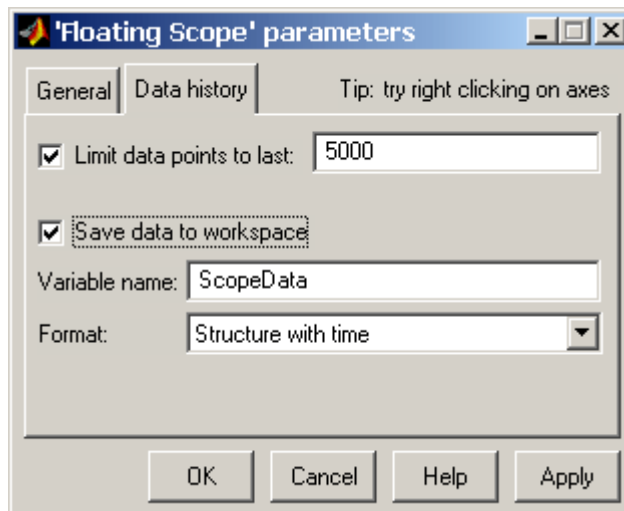


Рис. 4.23. Вкладка Data history

10.2.2. Цифровой дисплей Display

Назначение:

Отображает значение сигнала в виде числа.

Параметры:

- Format – формат отображения данных. Параметр Format может принимать следующие значения:

1. short – 5 значащих десятичных цифр.
2. long – 15 значащих десятичных цифр.
3. short_e – 5 значащих десятичных цифр и 3 символа степени десяти.
4. long_e – 15 значащих десятичных цифр и 3 символа степени десяти.
5. bank – «денежный» формат. Формат с фиксированной точкой и двумя десятичными цифрами в дробной части числа.

- Decimation – кратность отображения входного сигнала. При Decimation = 1 отображается каждое значение входного сигнала, при Decimation = 2 отображается каждое второе значение, при Decimation = 3 – каждое третье значение и т.д.
- Sample time – шаг модельного времени. Определяет дискретность отображения данных.
- Floating display (флажок) – перевод блока в «свободный» режим. В данном режиме входной порт блока отсутствует, а выбор сигнала для отображения выполняется щелчком левой клавиши мыши на соответствующей линии связи. В этом режиме для параметра расчета Signal storage reuse должно быть установлено значение off (вкладка Advanced в окне диалога Simulation parameters...).

На рис. 4.24 показано применение блока **Display** с использованием различных вариантов параметра **Format**.

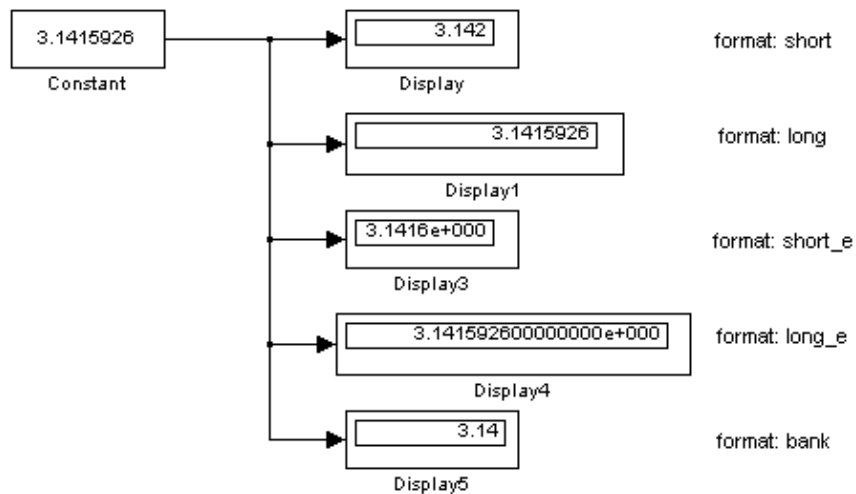


Рис. 4.24. Применение блока **Display** с использованием различных вариантов параметра **Format**

10.3. Continuous – аналоговые блоки

10.3.1. Интегрирующий блок Integrator

Назначение:

Выполняет интегрирование входного сигнала.

Параметры:

- External reset – внешний сброс. Тип внешнего управляющего сигнала, обеспечивающего сброс интегратора к начальному состоянию. Выбирается из списка:

1. none – нет (сброс не выполняется),
2. rising – нарастающий сигнал (передний фронт сигнала),
3. falling – спадающий сигнал (задний фронт сигнала),
4. either – нарастающий либо спадающий сигнал,
5. level – ненулевой сигнал (сброс выполняется, если сигнал на управляющем входе становится не равным нулю).

В том случае, если выбран какой-либо (но не none) тип управляющего сигнала, то на изображении блока появляется дополнительный управляющий вход. Рядом с дополнительным входом будет показано условное обозначение управляющего сигнала.

- Initial condition source – источник начального значения выходного сигнала. Выбирается из списка:

1. internal – внутренний;
2. external – внешний. В этом случае на изображении блока появляется дополнительный вход, обозначенный x_0 , на который необходимо подать сигнал, задающий начальное значение выходного сигнала интегратора.

- Initial condition – начальное условие. Установка начального значения выходного сигнала интегратора. Параметр доступен, если выбран внутренний источник начального значения выходного сигнала.
- Limit output (флажок) – использование ограничения выходного сигнала.

- Upper saturation limit – верхний уровень ограничения выходного сигнала. Может быть задан как числом, так и символьной последовательностью inf, то есть $+\infty$.
- Lower saturation limit – нижний уровень ограничения выходного сигнала. Может быть задан как числом, так и символьной последовательностью inf, то есть $-\infty$.
- Show saturation port – управляет отображением порта, выводящего сигнал, свидетельствующий о выходе интегратора на ограничение. Выходной сигнал данного порта может принимать следующие значения:
 1. ноль, если интегратор не находится на ограничении;
 2. +1, если выходной сигнал интегратора достиг верхнего ограничивающего предела;
 3. -1, если выходной сигнал интегратора достиг нижнего ограничивающего предела.
- Show state port (флажок) – отобразить/скрыть порт состояния блока. Данный порт используется в том случае, если выходной сигнал интегратора требуется подать в качестве сигнала обратной связи этого же интегратора. Например, при установке начальных условий через внешний порт или при сбросе интегратора через порт сброса. Выходной сигнал с этого порта может использоваться также для организации взаимодействия с управляемой подсистемой.
- Absolute tolerance – абсолютная погрешность.
- На рис. 4.25 показан пример работы интегратора при подаче на его вход ступенчатого сигнала. Начальное условие принято равным нулю.

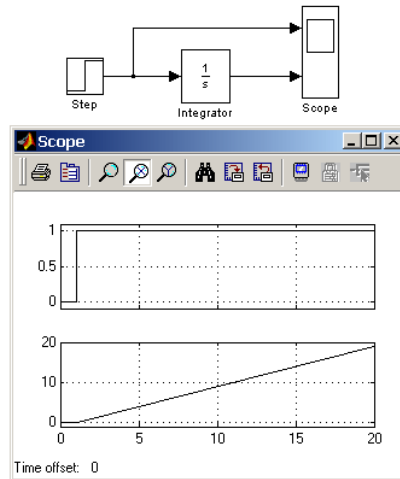


Рис. 4.25. Интегрирование ступенчатого сигнала

Пример на рис. 4.25 отличается от предыдущего подачей начального значения через внешний порт. Начальное значение выходного сигнала в данном примере задано равным 10.

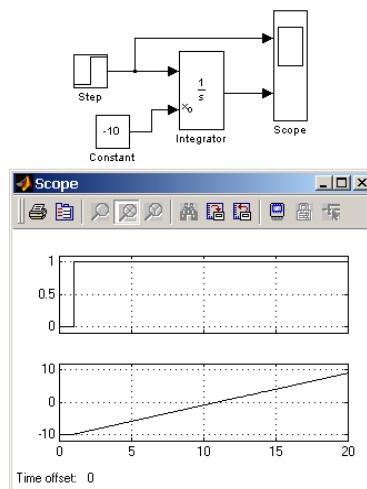


Рис. 4.26. Интегрирование ступенчатого сигнала с установкой начального значения выходного сигнала

10.3.2. Блок фиксированной задержки сигнала Transport Delay

Назначение:

Обеспечивает задержку входного сигнала на заданное время.

Параметры:

1. Time Delay – время задержки сигнала (неотрицательное значение).
2. Initial input – начальное значение выходного сигнала.
3. Buffer size – размер памяти, выделяемой для хранения задержанного сигнала. Задается в байтах числом, кратным 8 (по умолчанию 1024).
4. Pade order (for linearization) – порядок ряда Паде, используемого при аппроксимации выходного сигнала. Задается целым положительным числом.

При выполнении моделирования значение сигнала и соответствующее ему модельное время сохраняются во внутреннем буфере блока Transport Delay. По истечении времени задержки значение сигнала извлекается из буфера и передается на выход блока. В том случае, если шаги модельного времени не совпадают со значениями моментов времени для записанного в буфер сигнала, блок Transport Delay выполняет аппроксимацию выходного сигнала.

В том случае, если начального значения объема памяти буфера не хватит для хранения задержанного сигнала, Simulink автоматически выделит дополнительную память. После завершения моделирования в командном окне MATLAB появится сообщение с указанием нужного размера буфера. На рис. 4.27 показан пример использования блока Transport Delay для задержки прямоугольного сигнала на 0,5 с.

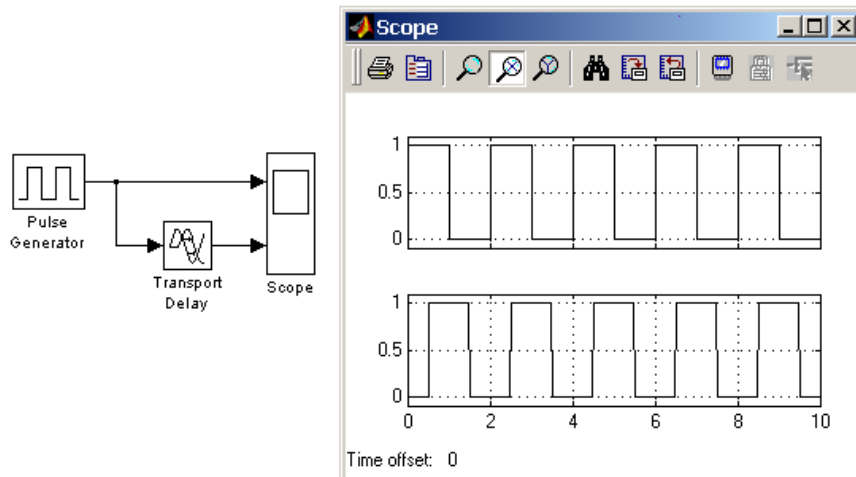


Рис. 4.27. Пример использования блока Transport Delay для задержки сигнала

10.3.3. Блок управляемой задержки сигнала Variable Transport Delay

Назначение:

Выполняет задержку входного сигнала, заданную величиной сигнала управления.

Параметры:

1. Maximum delay – максимальное значение времени задержки сигнала (не отрицательное значение).
2. Initial input – начальное значение выходного сигнала.
3. Buffer size – размер памяти, выделяемой для хранения задержанного сигнала. Задается в байтах числом, кратным 8 (по умолчанию 1024).
4. Pade order (for linearization) – порядок ряда Паде, используемого при аппроксимации выходного сигнала. Задается целым положительным числом.

Блок управляемой задержки Variable Transport Delay работает аналогично блоку постоянной задержки сигнала Transport Delay.

В том случае, если значение управляющего сигнала, задающего величину задержки, превышает значение, заданное параметром Maximum delay, то задержка выполняется на величину Maximum delay.

На рис. 4.28 показан пример использования блока Variable Transport Delay. Величина времени задержки сигнала изменяется от 0,5 до 1 с в момент времени равный 5 с.

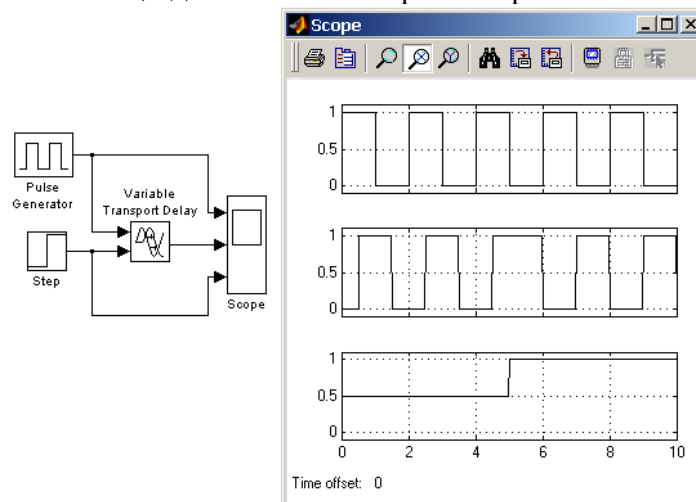


Рис. 4.28. Пример использования блока Variable Transport Delay

10.4. Nonlinear – нелинейные блоки

10.4.1. Блок ограничения Saturation

Назначение:

Выполняет ограничение величины сигнала.

Параметры:

1. Upper limit – верхний порог ограничения.
2. Lower limit – нижний порог ограничения.

3. Treat as gain when linearizing (флажок) – трактовать как усилитель с коэффициентом передачи равным 1 при линеаризации.

Выходной сигнал блока равен входному, если его величина не выходит за порог ограничения. По достижении входным сигналом уровня ограничения выходной сигнал блока перестает изменяться и остается равным порогу. На рис. 4.29 показан пример использования блока для ограничения синусоидального сигнала. На рисунке приводятся временные диаграммы сигналов и зависимость выходного сигнала блока от входного.

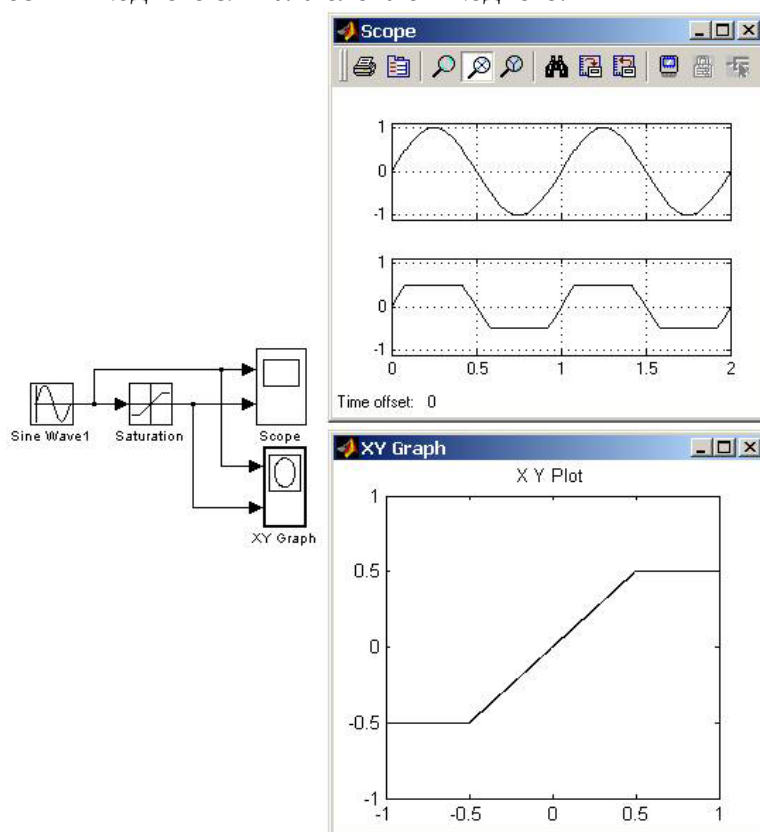


Рис. 4.29. Пример использования блока Saturation

10.4.2. Блок переключателя Switch

Назначение:

Выполняет переключение входных сигналов по сигналу управления.

Параметры:

Threshold – порог управляющего сигнала.

Блок работает следующим образом. Если сигнал управления, подаваемый на средний вход меньше, чем величина порогового значения Threshold, то на выход блока проходит сигнал с первого (верхнего) входа. Если сигнал управления превысит пороговое значение, то на выход блока будет поступать сигнал со второго (нижнего) входа.

На рис. 4.30 показан пример работы блока Switch. В том случае, когда сигнал на управляющем входе ключа равен 1, на выход блока проходит гармонический сигнал, если же управляющий сигнал равен нулю, то на выход проходит сигнал нулевого уровня от блока Ground. Пороговое значение управляющего сигнала задано равным 0,5.

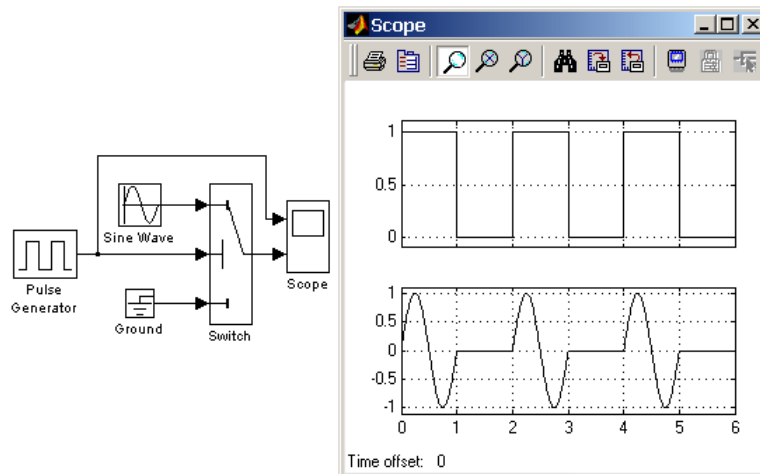


Рис. 4.30. Применение переключателя Switch

10.4.3. Блок ручного переключателя Manual Switch

Назначение:

Выполняет переключение входных сигналов по команде пользователя.

Параметры:

Нет.

Командой на переключение является двойной щелчок левой клавишей мыши на изображении блока. При этом изображение блока изменяется, показывая, какой входной сигнал в данный момент проходит на выход блока. Переключение блока можно выполнять как до начала моделирования, так и в процессе расчета.

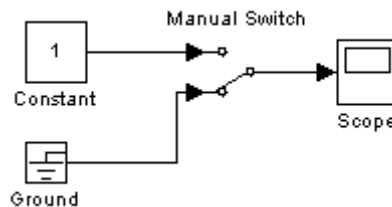


Рис. 4.31. Пример использования блока Manual Switch

10.5. Math – блоки математических операций

10.5.1. Блок вычисления суммы Sum

Назначение:

Выполняет вычисление суммы текущих значений сигналов.

Параметры:

1. Icon shape – форма блока. Выбирается из списка: round – окружность, rectangular – прямоугольник.
2. List of sign – список знаков. В списке можно использовать следующие знаки: + (плюс), - (минус) и | (разделитель знаков).
3. Saturate on integer overflow (флажок) – подавлять переполнение целого. При установленном флажке ограничение сигналов целого типа выполняется корректно.

Количество входов и операция (сложение или вычитание) определяется списком знаков параметра List of sign, при этом метки входов обозначаются соответствующими знаками. В параметре List of sign можно также указать число входов блока. В этом случае все входы будут суммирующими.

Если количество входов блока превышает 3, то удобнее использовать блок Sum прямоугольной формы.

Блок может использоваться для суммирования скалярных, векторных или матричных сигналов. Типы суммируемых сигналов должны совпадать. Нельзя, например, подать на один и тот же суммирующий блок сигналы целого и действительного типов.

Если количество входов блока больше, чем один, то блок выполняет поэлементные операции над векторными и матричными сигналами. При этом количество элементов в матрице или векторе должно быть одинаковым.

Если в качестве списка знаков указать цифру 1 (один вход), то блок можно использовать для определения суммы элементов вектора.

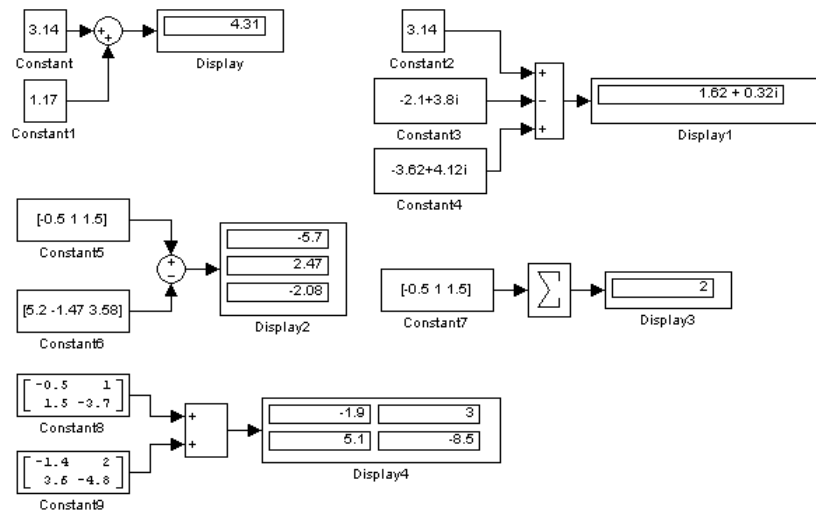


Рис. 4.32. Примеры использования блока Sum

10.5.2. Усилители Gain и Matrix Gain

Назначение:

Выполняют умножение входного сигнала на постоянный коэффициент.

Параметры:

1. Gain – коэффициент усиления.
2. Multiplication – способ выполнения операции. Может принимать значения (из списка): Element-wise $K*u$ – Поэлементный; Matrix $K*u$ – матричный (коэффициент усиления является левосторонним операндом); Matrix $u*K$ – матричный (коэффициент усиления является правосторонним операндом).
3. Saturate on integer overflow (флажок) – подавлять переполнение целого. При установленном флажке ограничение сигналов целого типа выполняется корректно.

Блоки усилителей Gain и Matrix Gain есть один и тот же блок, но с разными начальными установками параметра Multiplication.

Параметр блока Gain может быть положительным или отрицательным числом, как больше, так и меньше 1. Коэффициент усиления можно задавать в виде скаляра, матрицы или вектора, а также в виде вычисляемого выражения.

В том случае если параметр Multiplication задан как Element-wise $K*u$, то блок выполняет операцию умножения на заданный коэффициент скалярного сигнала или каждого элемента векторного сигнала. В противном случае блок выполняет операцию матричного умножения сигнала на коэффициент, заданный матрицей.

По умолчанию коэффициент усиления является действительным числом типа double.

Для операции поэлементного усиления входной сигнал может быть скалярным, векторным или матричным любого типа, за исключением логического (boolean). Элементы вектора должны иметь одинаковый тип сигнала. Выходной сигнал блока будет иметь тот же самый тип, что и входной сигнал.

Параметр блока Gain может быть скаляром, вектором или матрицей любого типа, за исключением логического (boolean).

При вычислении выходного сигнала блок Gain использует следующие правила:

- Если входной сигнал действительного типа, а коэффициент усиления комплексный, то выходной сигнал будет комплексным.
- Если тип входного сигнала отличается от типа коэффициента усиления, то Simulink пытается выполнить приведение типа коэффициента усиления к типу входного сигнала. Если такое приведение невозможно, расчет будет остановлен с выводом сообщения об ошибке. Такая ситуация может возникнуть, например, если входной сигнал есть беззнаковое целое (uint8), а параметр Gain задан отрицательным числом.

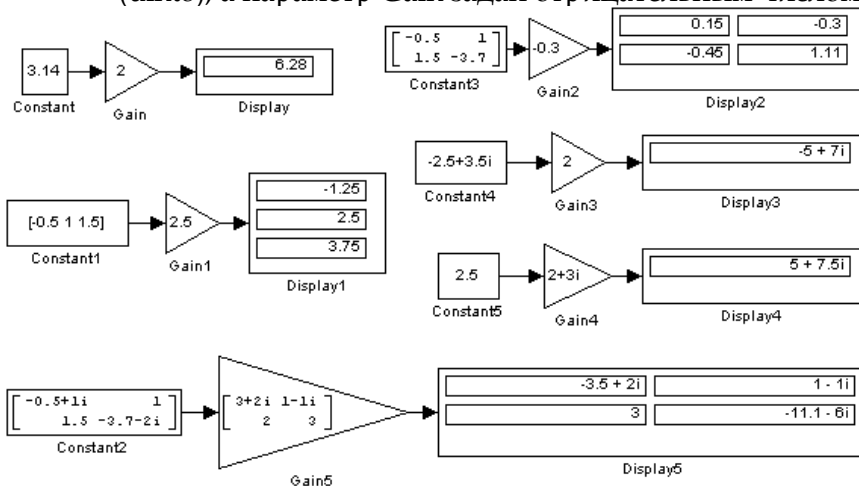


Рис. 4.33. Примеры использования блока Gain

Для операций матричного усиления (матричного умножения входного сигнала на заданный коэффициент) входной сигнал и коэффициент усиления должны быть скалярными,

векторными или матричными значениями комплексного либо действительного типа `single` или `double`.

10.5.3. Блок вычисления операции отношения **Relational Operator**

Назначение:

Блок сравнивает текущие значения входных сигналов.

Параметры:

Relational Operator – тип операции отношения (выбирается из списка):

- `==` – Тождественно равно;
- `~=` – Не равно;
- `<` – Меньше;
- `<=` – Меньше или равно;
- `>=` – Больше или равно;
- `>` – Больше.

В операции отношения первым операндом является сигнал, подаваемый на первый (верхний) вход блока, а вторым операндом – сигнал, подаваемый на второй (нижний) вход. Выходным сигналом блока является 1, если результат вычисления операции отношения есть «ИСТИНА», и 0, если результат – «ЛОЖЬ».

Входные сигналы блока могут быть скалярными, векторными или матричными. Если оба входных сигнала – векторы или матрицы, то блок выполняет поэлементную операцию сравнения, при этом размерность входных сигналов должна совпадать. Если один из входных сигналов – вектор или матрица, а другой входной сигнал – скаляр, то блок выполняет сравнение скалярного входного сигнала с каждым элементом массива. Размерность выходного сигнала в этом случае будет определяться размерностью векторного или матричного сигнала, подаваемого на один из входов.

Для операций `==` (тождественно равно) и `~=` (не равно) допускается использовать комплексные входные сигналы.

Входные сигналы также могут быть логического типа (boolean).

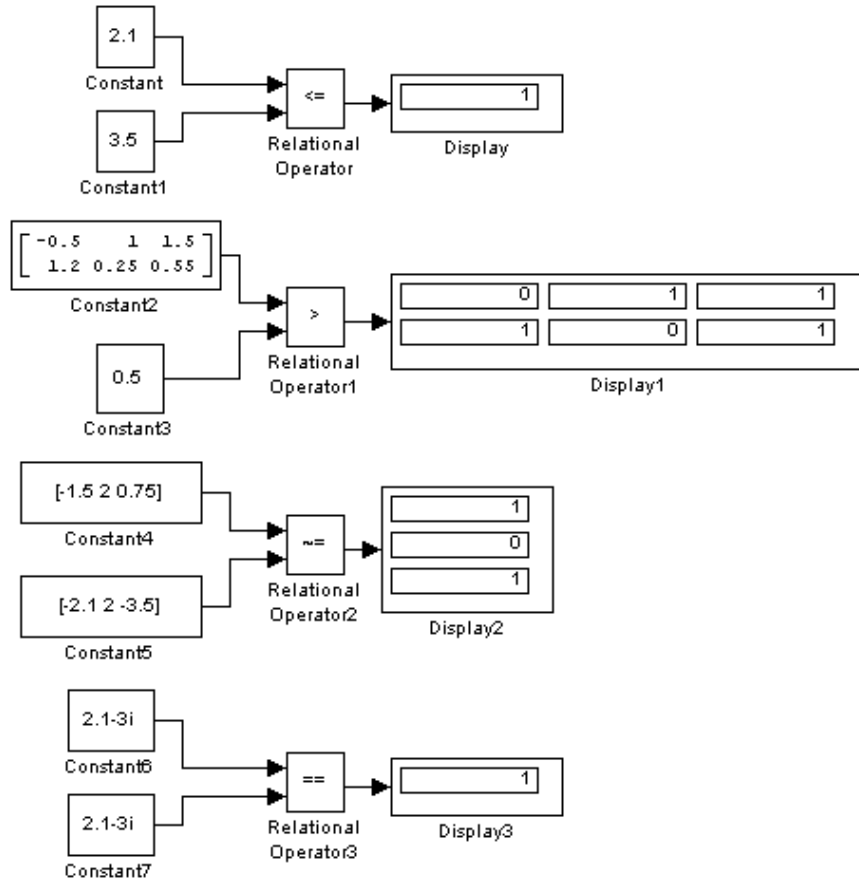


Рис. 4.34. Примеры использования блока Relational Operator

10.6. Signal&Systems – блоки преобразования сигналов и вспомогательные блоки

10.6.1. Мультиплексор (смеситель) Mux

Назначение:

Объединяет входные сигналы в вектор.

Параметры:

1. **Number of Inputs** – количество входов.
2. **Display option** – способ отображения. Выбирается из списка:
 - **bar** – вертикальный узкий прямоугольник черного цвета;
 - **signals** – прямоугольник с белым фоном и отображением меток входных сигналов;
 - **none** – прямоугольник с белым фоном без отображения меток входных сигналов.

Входные сигналы блока могут быть скалярными и (или) векторными.

Если среди входных сигналов есть векторы, то количество входов можно задавать как вектор с указанием числа элементов каждого вектора. Например, выражение [2 3 1] определяет три входных сигнала: первый сигнал – вектор из двух элементов, второй сигнал – вектор из трех элементов, и последний сигнал – скаляр. Если размерность входного вектора не совпадает с указанной в параметре Number of Inputs, после начала расчета Simulink выдаст сообщение об ошибке. Размерность входного вектора можно задавать как -1 (минус один). В этом случае размерность входного вектора может быть любой.

Параметр Number of Inputs можно задавать также в виде списка меток сигналов, например: Vector1, Vector2, Scalar. В этом случае метки сигналов будут отображаться рядом с соответствующими соединительными линиями.

Сигналы, подаваемые на входы блока, должны быть одного типа (действительного или комплексного).

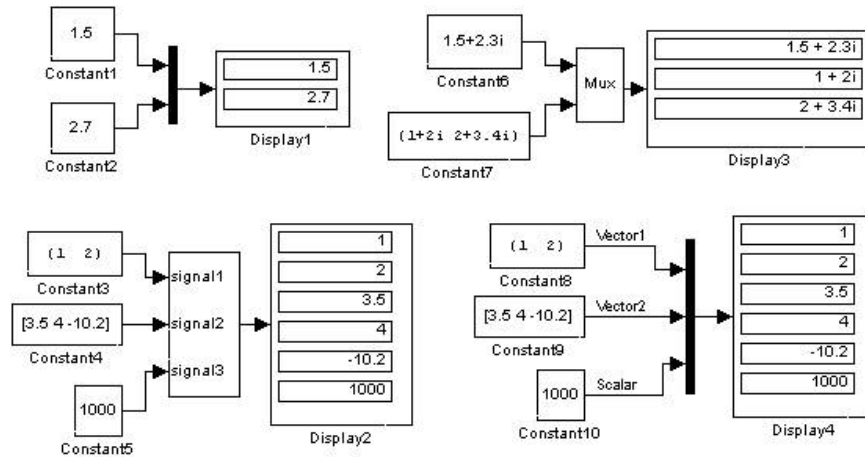


Рис. 4.35. Примеры использования блока Mux

10.6.2. Демультимплексор (разделитель) Demux

Назначение:

Разделяет входной векторный сигнал на отдельные составляющие.

Параметры:

1. Number of Outputs – количество выходов.
2. Bus Selection Mode (флажок) – режим разделения векторных сигналов.

Входными сигналами в обычном режиме является вектор, сформированный любым способом. Выходными сигналами являются скаляры или векторы, количество которых и размерность определяется параметром Number of Outputs и размерностью входного вектора.

Если количество выходов P (значение параметра Number of Outputs) равно размерности входного сигнала N , то блок выполняет разделение входного вектора на отдельные элементы.

Если количество выходов P меньше, чем размерность входного сигнала N , то размерность первых $P-1$ выходных сигналов равна отношению N/P , округленному до ближай-

шего большего числа, а размерность последнего выходного сигнала равна разности между размерностью входного сигнала и суммой размерностей первых $P-1$ выходов. Например, если размерность входного сигнала равна 8, а количество выходов равно 3, то первые два выходных вектора будут иметь размерность $\text{ceil}(8/3) = 3$, а последний выходной вектор будет иметь размерность $8 - (3 + 3) = 2$.

Параметр Number of Outputs может быть задан также с помощью вектора, определяющего размерность каждого выходного сигнала. Например, выражение $[2 \ 3 \ 1]$ определяет три выходных сигнала: первый сигнал – вектор из двух элементов, второй сигнал – вектор из трех элементов, и последний сигнал – скаляр. Размерность можно также задавать как -1 (минус один). В этом случае размерность соответствующего выходного сигнала определяется как разность между размерностью входного вектора и суммой размерностей заданных выходных сигналов. Например, если размерность входного вектора равна 6, а параметр Number of Outputs задан выражением $[1 \ -1 \ 3]$, то второй выходной сигнал будет иметь размерность $6 - (3 + 1) = 2$.

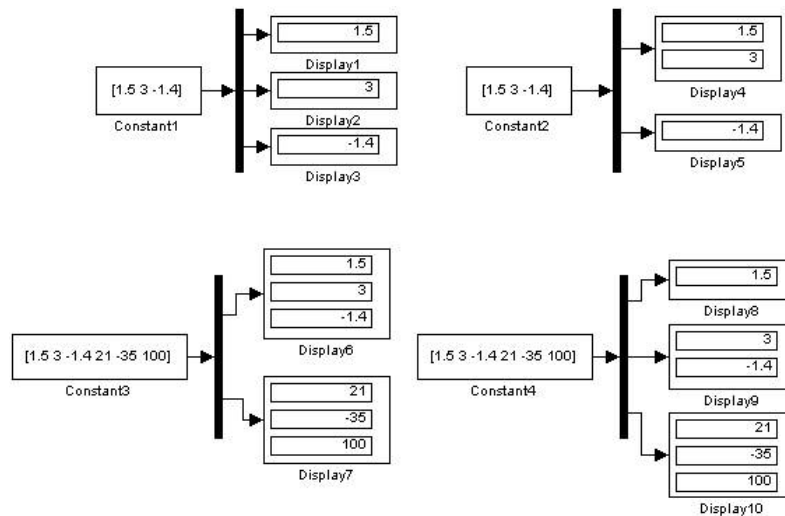


Рис. 4.36. Примеры использования блока Demux

В режиме Bus Selection Mode блок Demux работает не с отдельными элементами векторов, а с векторными сигналами в целом. Входной сигнал в этом режиме должен быть сформирован блоком Mux или другим блоком Demux. Параметр Number of Outputs в этом случае задается в виде скаляра, определяющего количество выходных сигналов, либо в виде вектора, каждый элемент которого определяет количество векторных сигналов в данном выходном сигнале. Например, при входном сигнале, состоящем из трех векторов, параметр Number of Outputs, заданный вектором [2 1], определит два выходных сигнала, первый из которых будет содержать два векторных сигнала, а второй – один.

10.7. Function & Tables – блоки функций и таблиц

10.7.1. Блок задания функции Fcn

Назначение:

Задаёт выражение в стиле языка программирования (C++).

Параметры:

Expression – выражение, используемое блоком для вычисления выходного сигнала на основании входного. Это выражение составляется по правилам, принятым для описания функций на языке C.

В выражении можно использовать следующие компоненты:

1. Входной сигнал. Входной сигнал в выражении обозначается **u**, если он является скаляром. Если входной сигнал – вектор, необходимо указывать номер элемента вектора в круглых скобках. Например, **u(1)** и **u(3)** – первый и третий элементы входного вектора.
2. Константы.
3. Арифметические операторы (+ - * /).
4. Операторы отношения (= != > < >= <=).
5. Логические операторы (&& | |!).
6. Круглые скобки.

7. Математические функции: `abs`, `acos`, `asin`, `atan`, `atan2`, `ceil`, `cos`, `cosh`, `exp`, `fabs`, `floor`, `hypot`, `ln`, `log`, `log10`, `pow`, `power`, `rem`, `sgn`, `sin`, `sinh`, `sqrt`, `tan`, и `tanh`.
8. Переменные из рабочей области. Если переменная рабочей области является массивом, то ее элементы должны указываться с помощью индексов в круглых скобках. Например, `A(1,1)` – первый элемент матрицы `A`.

Операторы отношения и логические операторы возвращают значения в виде логического нуля (`FALSE`) или логической единицы (`TRUE`).

Операторы, допускаемые к использованию в выражении, имеют следующий приоритет (в порядке убывания):

1. `()`
2. `+` `-` (унарные)
3. Возведение в степень
4. `!`
5. `/`
6. `+` `-` (бинарные)
7. `>` `<` `<=` `>=`
8. `=` `!=`
9. `&&`
10. `|` `|`

Блок не поддерживает матричные и векторные операции. Выходной сигнал блока всегда – скаляр.

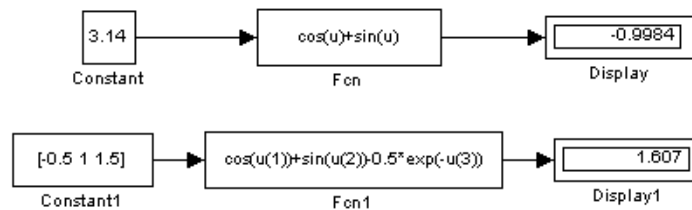


Рис. 4.37. Примеры использования блока Fcn

10.7.2. Блок задания функции MATLAB Fcn

Назначение:

Задаёт выражение в стиле языка программирования **MATLAB**.

Параметры:

1. MATLAB function – выражение на языке MATLAB.
2. Output dimensions – размерность выходного сигнала. Значение параметра -1 (минус один) предписывает блоку определять размерность автоматически.
3. Output signal type – тип выходного сигнала. Выбирается из списка:
 - real – действительный сигнал;
 - complex – комплексный сигнал;
 - auto – автоматическое определение типа сигнала.
4. Collapse 2-D results to 1-D – Преобразование двумерного выходного сигнала к одномерному.

Входной сигнал в выражении обозначается *u*, если он является скаляром. Если входной сигнал – вектор, необходимо указывать номер элемента вектора в круглых скобках. Например, *u(1)* и *u(3)* – первый и третий элементы входного вектора. Если выражение состоит из одной функции, то ее можно задать без указания параметров. Выражение может содержать также собственные функции пользователя, написанные на языке MATLAB и оформленные в виде *m*-файлов. Имя *m*-файла не должно совпадать с именем модели (*mdl*-файлом).

Рис. 4.38 демонстрирует применение блока MATLAB Fcn. В примере используется функция *My_Matlab_Fcn_1*, вычисляющая сумму и произведение двух элементов входного вектора. Текст функции (файл *My_Matlab_Fcn_1.m*) приведен ниже:

```
function y=My_Matlab_Fcn_1(x,k);  
y(1)=x*k;  
y(2)=x + k;
```

Выражение для вызова функции, заданное параметром MATLAB function, имеет вид: *My_Matlab_Fcn_1(u(1),u(2))*.

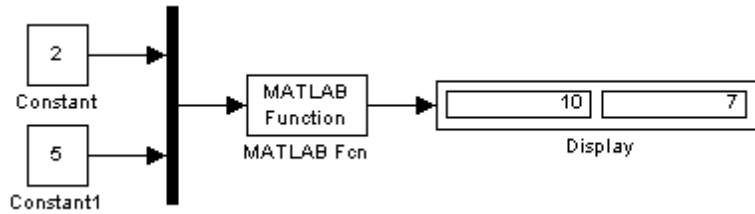


Рис. 4.38. Примеры использования блока MATLAB Fcn

10.8. Этапы моделирования

Процесс расчета модели выполняется Simulink в несколько этапов. На первом этапе выполняется инициализация модели: подключение библиотечных блоков к модели, определение размерностей сигналов, типов данных, величин шагов модельного времени, оценка параметров блоков, а также определяется порядок выполнения блоков и происходит выделение памяти для проведения расчета. Затем Simulink начинает цикл моделирования. На каждом цикле моделирования (временном шаге) происходит расчет блоков в порядке, определенном на этапе инициализации. Для каждого блока Simulink вызывает функции, которые вычисляют переменные состояния блока x , производные переменных состояния и выходы y в течение текущего шага модельного времени. Этот процесс продолжается, пока моделирование не будет завершено. На рис. 4.39 показана диаграмма, иллюстрирующая этот процесс.

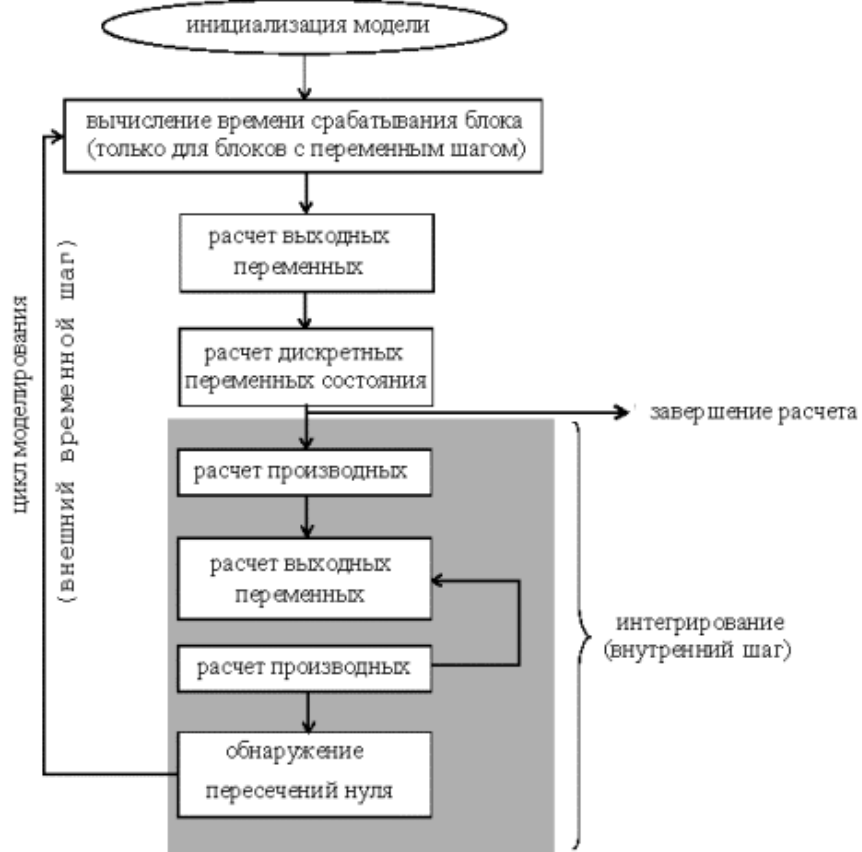


Рис. 4.39. Процесс моделирования

Глава 5. Управление модельным временем

§1. Виды представления времени в модели

С построенной имитационной моделью проводятся имитационные эксперименты, которые фактически представляют собой наблюдение за поведением экономической системы в течение некоторого промежутка времени. Эффективность экономической системы с помощью модели оценивается путем наблюдения за ней во времени. Характерной особенностью большинства практических задач является то, что скорость протекания рассматриваемых экономических процессов значительно ниже скорости реализации модельного эксперимента. Например, если моделируется работа супермаркета в течение месяца, вряд ли кому-то придет в голову воспроизводить этот процесс в модели в таком же масштабе времени. С другой стороны, даже те имитационные эксперименты, в которых временные параметры функционирования системы не учитываются, требуют для своей реализации определенных затрат времени работы компьютера.

В связи с этим при разработке практически любой имитационной модели и планировании проведения модельных экспериментов необходимо соотносить между собой три представления времени:

- реальное, в котором происходит функционирование имитируемой системы;
- модельное (или, как его еще называют, системное), в масштабе которого организуется работа модели;
- машинное, отражающее затраты времени ЭВМ на проведение имитации.

С помощью механизма модельного времени решаются следующие задачи:

- Отображается переход моделируемой системы из одного состояния в другое.
- Производится синхронизация работы компонент модели.

- Изменяется масштаб времени «жизни» (функционирования) исследуемой системы.
- Производится управление ходом модельного эксперимента.
- Моделируется квазипараллельная реализация событий в модели (приставка «квази» в данном случае отражает последовательный характер обработки событий (процессов) в ИМ, которые в реальной системе возникают (протекают) одновременно).

Необходимость решения последней задачи связана с тем, что в распоряжении исследователя, как правило, находится однопроцессорная вычислительная система, а модель может содержать значительно большее число одновременно работающих подсистем. Поэтому действительно параллельная (одновременная) реализация всех компонент модели невозможна. Реализация квазипараллельной работы компонент модели является достаточно сложной технической задачей. Пакет Matlab позволяет избавиться от этой проблемы. В данном курсе вопросы, связанные с этой проблемой, не рассматриваются.

Существуют два метода реализации механизма модельного времени:

- моделирование с постоянным шагом;
- моделирование по особым состояниям.

§2. Изменение времени с постоянным шагом

При использовании данного метода отсчет системного времени ведется через фиксированные, выбранные исследователем интервалы времени. События в модели считаются наступившими в момент окончания этого интервала. Погрешность в измерении временных характеристик системы в этом случае зависит от величины шага моделирования ΔT .

Метод моделирования с постоянным шагом используется на практике, если:

1. События появляются регулярно, их распределение во времени достаточно равномерно.
2. Число событий велико и моменты их появления близки.
3. Моменты появления событий заранее определить невозможно.

Данный метод управления модельным временем достаточно просто реализуется в том случае, когда условия появления событий всех типов в модели можно представить как функцию времени (например, если моделируется система массового обслуживания).

Рассмотрим в качестве примера систему массового обслуживания, процессы которой мы хотим моделировать.

Процесс функционирования такой системы можно рассматривать как последовательную смену ее состояний. Пусть, например, в одноканальной системе массового обслуживания происходит процесс обслуживания поступающих заявок (рис. 5.1).

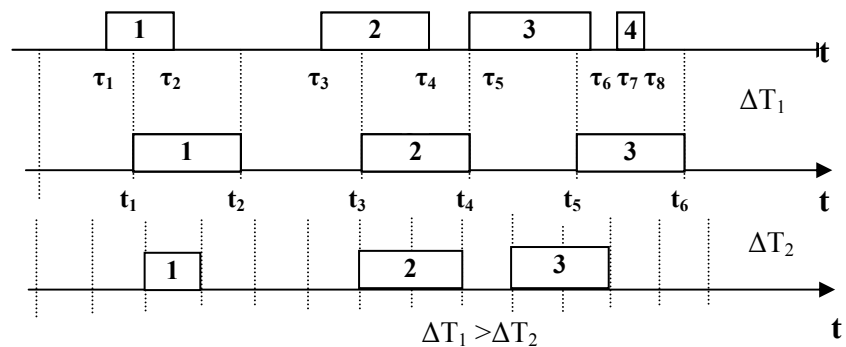


Рис. 5.1. Моделирование способом ΔT

Введем следующие обозначения:

- τ_1 – момент начала обслуживания 1-й заявки;
- τ_2 – момент конца обслуживания 1-й заявки;
- τ_3 – момент начала обслуживания 2-й заявки;
- τ_4 – момент конца обслуживания 2-й заявки;

t_5 – момент начала обслуживания 3-й заявки;

t_6 – момент конца обслуживания 3-й заявки;

t_7 – момент начала обслуживания 4-й заявки;

t_8 – момент конца обслуживания 4-й заявки.

Выберем шаг ΔT и будем анализировать состояние системы через промежутки времени t_1, t_2, \dots , отстоящие друг от друга на ΔT . Этот способ иногда называют способом ΔT .

В момент t_1 будет обнаружено, что в системе началось обслуживание 1-й заявки. В момент $t_2 = t_1 + \Delta T$ будет установлено, что обслуживание 1-й заявки завершено. В момент $t_3 = t_2 + \Delta T$ будет обнаружено, что в системе началось обслуживание 2-й заявки. В момент $t_4 = t_3 + \Delta T$ будет установлено, что обслуживание 2-й заявки завершено. В момент $t_5 = t_4 + \Delta T$ будет обнаружено, что в системе началось обслуживание 3-й заявки. В момент $t_6 = t_5 + \Delta T$ будет установлено, что обслуживание 3-й заявки завершено. Факт поступления 4-й заявки и факт окончания ее обслуживания не будут обнаружены.

Эту логику наблюдения за реальной системой мы переносим из реального времени поведения системы в модельное время.

Для предотвращения потерь информации и повышения точности работы модели нужно уменьшить шаг ΔT . При малом ΔT можно достаточно точно описать процесс функционирования системы.

Однако способ ΔT является весьма неэкономичным с точки зрения расхода машинного времени. Достоинство способа состоит в том, что он позволяет моделировать любые процессы: детерминированные, непрерывные, случайные, с зависимыми или независимыми событиями и т.п.

Выбор ΔT – задача очень важная и нелегкая. Необходимо:

1. ΔT принимать равной средней интенсивности возникновения событий различных типов.
2. ΔT выбирать равной среднему интервалу между наиболее частыми (или наиболее важными) событиями.

§3. Продвижение времени по особым состояниям

При моделировании по особым состояниям системное время каждый раз изменяется на величину, строго соответствующую интервалу времени до момента наступления очередного события. В этом случае события обрабатываются в порядке их наступления, а одновременно наступившими считаются только те, которые являются одновременными в действительности.

Для реализации моделирования по особым состояниям требуется разработка специальной процедуры планирования событий (так называемого календаря событий).

Если известен закон распределения интервалов между событиями, то такое прогнозирование труда не составляет: достаточно к текущему значению модельного времени добавить величину интервала, полученную с помощью соответствующего датчика.

Пусть, например, при моделировании системы массового обслуживания очередные заявки поступают в случайные моменты времени, но по известному закону, допустим показательному (именно так часто бывает на практике) с параметрами: λ – интенсивность потока заявок и T – среднее время между соседними заявками. Иллюстрация к такой ситуации приведена на рис. 5.2 ($t_1 - t_4$ – моменты формирования заявок для системы массового обслуживания; $\Delta T_1 \dots \Delta T_4$ – случайные интервалы времени, имеющие показательный закон распределения СВ).

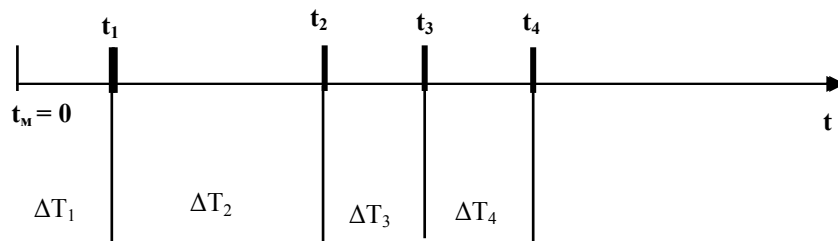


Рис. 5.2. Изменение модельного времени по особым состояниям

При моделировании таким методом сложности возникают, если имеет место несколько взаимосвязанных событий.

Моделирование по особым состояниям целесообразно использовать, если:

- события распределяются во времени неравномерно или интервалы между ними велики;
- предъявляются повышенные требования к точности определения взаимного положения событий во времени;
- необходимо учитывать наличие одновременных событий.

Дополнительное достоинство метода заключается в том, что он позволяет экономить машинное время, особенно при моделировании систем периодического действия, в которых события длительное время могут не наступать.

§4. Моделирование параллельных процессов

Практически любая более или менее сложная система имеет в своем составе компоненты, работающие одновременно, или, как принято говорить, параллельно. Примерами являются одновременное обслуживание клиентов, например, в парикмахерской, работа по обслуживанию и одновременно устранению неисправности в работе, например, одного из кассовых аппаратов, и т.д.

Итак, если в составе системы имеются компоненты (подсистемы), выполняющие свои функции одновременно, то можно утверждать, что в такой системе существуют параллельные процессы. Параллельно работающие подсистемы могут взаимодействовать самым различным образом либо вообще работать независимо друг от друга. Способ взаимодействия подсистем определяет вид параллельных процессов, протекающих в системе. В свою очередь, вид моделируемых процессов влияет на выбор метода их имитации.

Виды параллельных процессов

Асинхронный параллельный процесс – такой процесс, состояние которого не зависит от состояния другого параллельного процесса (ПП).

Пример асинхронных ПП, протекающих в рамках одной системы: подготовка и проведение рекламной кампании фирмой и работа сборочного конвейера; пример из области вычислительной техники: выполнение вычислений процессором и вывод информации на печать.

Синхронный ПП – такой процесс, состояние которого зависит от состояния взаимодействующих с ним ПП.

Пример синхронного ПП: работа торговой организации и доставка товара со склада (нет товара – нет торговли).

Независимый ПП – процесс, который не является подчиненным ни для одного из процессов. Пример независимого ПП: процессы обслуживания покупателей в кассах супермаркета.

Методы описания параллельных процессов

В пакете Matlab имеется собственный язык моделирования, позволяющий моделировать параллельные процессы.

Рассмотрим механизм реализации ПП на основе транзактов.

В этом случае под событием понимается любое перемещение транзакта по системе, а также изменение его состояния (обслуживается, заблокирован и т.д.).

Событие, связанное с данным транзактом, может храниться в одном из следующих списков:

- **Список текущих событий.** В этом списке находятся события, время наступления которых меньше или равно текущему модельному времени. События с «меньшим» временем связаны с перемещением тех транзактов, которые должны были начать двигаться, но были заблокированы.
- **Список будущих событий.** Этот список содержит события, время наступления которых больше текущего модельного времени, то есть события, которые должны произойти в будущем (условия наступления которых уже определены – например, известно, что транзакт будет обслуживаться некоторым устройством 10 единиц времени).

- **Список прерываний.** Данный список содержит события, связанные с возобновлением обработки прерванных транзактов. События из этого списка выбираются в том случае, если сняты условия прерывания.

Рассмотрим использование двух первых списков событий в динамике, при моделировании параллельных процессов.

В списке текущих событий транзакты расположены в порядке убывания приоритета соответствующих событий; при равных приоритетах – в порядке поступления в список.

Каждое событие (транзакт) в списке текущих событий может находиться либо в активном состоянии, либо в состоянии задержки. Если событие активно, то соответствующий транзакт может быть продвинут по системе; если продвижение невозможно (например, из-за занятости устройства), то событие (и транзакт) переводится в состояние задержки.

Как только завершается обработка (продвижение) очередного активного транзакта, просматривается список задержанных транзактов и ряд из них переводится в активное состояние. Процедура повторяется до тех пор, пока в списке текущих событий не будут обработаны все активные события. После этого просматривается список будущих событий. Модельному времени присваивается значение, равное времени наступления ближайшего из этих событий. Данное событие заносится в список текущих событий. Затем просматриваются остальные события списка. Те из них, время которых равно текущему модельному времени, также переписываются в список текущих событий. Просмотр заканчивается, когда в списке остаются события, времена которых больше текущего модельного времени.

В качестве иллюстрации к изложенному рассмотрим небольшой пример. Пусть в систему поступают транзакты трех типов, каждый из которых обслуживается отдельным устройством. Известны законы поступления транзактов в систему и длительность их обслуживания. Таким образом, в системе существуют три параллельных независимых процесса (P1, P2, P3).

Временная диаграмма работы системы при обслуживании одного транзакта каждого типа показана на рис. 5.3.

На рисунке события, относящиеся к процессу P1, обозначены как C1i; относящиеся к P2 и к P3 – соответственно как C2i и C3i.

Для каждого процесса строится своя цепь событий, однако списки событий являются общими для всей модели. Формирование списков начинается с заполнения списка будущих событий. Как было отмечено выше, в этот список помещаются события, время наступления которых превышает текущее значение модельного времени. Очевидно, что на момент заполнения списка время наступления прогнозируемых событий должно быть известно. На первом шаге $t_m = 0$, и в список будущих событий заносятся события C11, C21, C31. Затем, событие с наименьшим временем наступления – C11 – переносится в список текущих событий; если одновременных с ним событий нет, то оно обрабатывается и исключается из списка текущих событий. После этого вновь корректируется список будущих событий и т.д., пока не истечет заданный интервал моделирования.

Динамика изменения списков текущих и будущих событий для рассмотренного примера отражена в таблице.

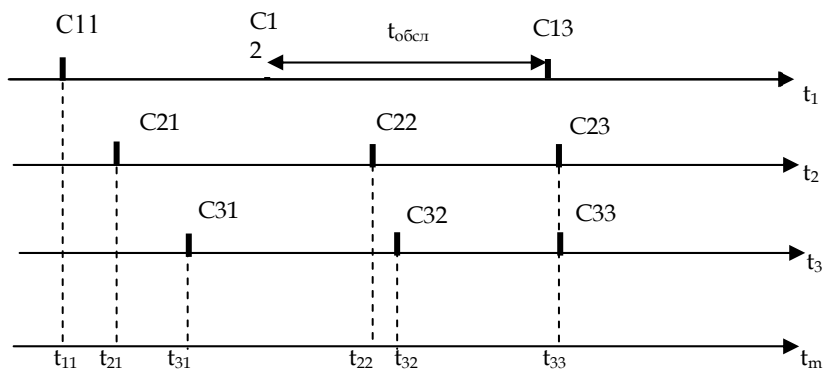


Рис. 5.3. Временная диаграмма параллельных процессов

Изменение списков событий

t	Список будущих событий	Список текущих событий
0	C11 C21 C31	0
t ₁₁	C21 C31 C12	C11
t ₂₁	C31 C12 C22	C21
t ₃₁	C12 C22 C32	C31
t ₁₂	C22 C32 C13	C12
t ₂₂	C32 C13 C23	C22
t ₃₂	C13 C23 C33	C32
t ₁₃	C23 C33	C32
t ₂₃		C23 C33

Применение сетевых моделей для описания параллельных процессов

Как будет показано на последующих занятиях, этапу программной реализации модели (т.е. ее описанию на одном из языков программирования) должен предшествовать так называемый этап алгоритмизации. Другими словами, прежде чем превратить имитационную модель в работающую компьютерную программу, ее создатель должен воспользоваться каким-то менее формальным и более наглядным средством описания логики работы будущей программы. Разумеется, это требование не является обязательным для всех разработчиков и для всех создаваемых моделей: при наличии достаточного опыта программа не очень сложной модели может быть написана сразу. Однако практика показывает, что человеческие возможности не безграничны, и при моделировании более сложных систем даже опытные разработчики бывают вынуждены немного «притормозить» на этапе алгоритмизации. Для описания логики работы модели могут быть использованы различные средства: либо русский язык (устный или письменный), либо традиционные схемы алгоритмов, либо какие-то другие «подручные» средства. Первые два варианта являются, как правило, наиболее знакомыми и наиболее часто используемыми. Однако, если вы попытаетесь описать в виде схемы алгоритма модель даже такой простой системы, которая использовалась в предыдущем примере, то это окажется напрасной тратой времени и сил. Прежде всего потому, что такие схемы совершенно не приспособлены для описания параллельных процессов.

Одним из наиболее элегантных и весьма распространенных средств описания параллельных процессов являются так называемые сети Петри.

§5. Управление модельным временем в MATLAB

Задача корректного управления модельным временем, то есть «временем жизни» моделируемой системы, возлагается на разработчика независимо от того, какие инструменты создания модели он использует. Принципиально отличие Simulink от универсальных средств программирования состоит в том, что логическая структура S-модели не зависит от способа управления модельным временем. Более того, исследователь может выбирать способ изменения модельного времени для каждого сеанса моделирования индивидуально.

Тем не менее и при использовании Simulink модельное время остается «спинным мозгом», согласующим работу всех компонент S-модели. Поэтому при подготовке каждого модельного эксперимента должны быть получены ответы на три вопроса:

- Какой способ изменения (приращения) времени целесообразно использовать (с переменным или постоянным шагом).
- С какой дискретностью следует изменять модельное время.
- Какое событие будет служить условием окончания эксперимента.

Выбор шага моделирования

Как было рассмотрено ранее, в практике имитационного моделирования применяются два основных способа изменения модельного времени – с постоянным шагом и по особым состояниям. При выборе одного из этих методов важное значение имеет тип моделируемой системы: для непрерывных систем (с непрерывным временем смены состояний) по умолчанию используется переменный шаг приращения времени, а для дискретных систем следует устанавливать постоянный (фиксированный) шаг. Но такой подход не всегда оправдан, поскольку при моделировании непрерывных систем бывает

удобнее определять очередное состояние системы как функцию времени, изменяющегося с заданной дискретностью. И наоборот, при моделировании дискретных систем величина очередного приращения времени зачастую определяется прогнозируемым моментом изменения состояния системы; причем смена состояний происходит, как правило, нерегулярно. Поэтому полезно знать, каким образом при разработке моделей дискретных систем можно заставить модельное время изменяться по особым состояниям.

Ранее мы рассматривали то, что в окне Simulink имеется меню Simulation (рис. 5.4).

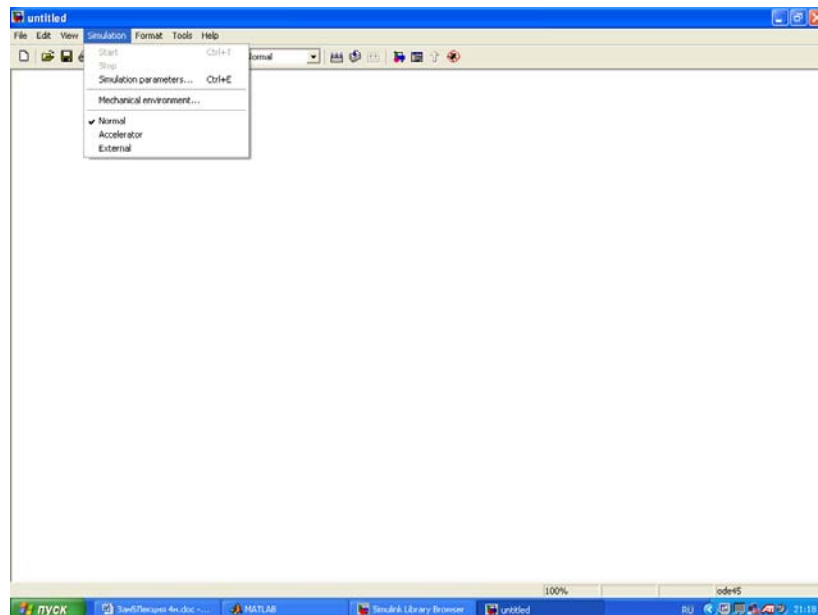


Рис. 5.4. Окно блок-диаграммы S-модели и вид меню Simulation

Данное меню играет основную роль при проведении исследования на модели. Посредством команд этого меню разработчик получает возможность не только динамически управлять сеансом моделирования, но и изменять многие важней-

шие параметры модели, такие как, например, способ изменения модельного времени и формат представления результатов моделирования.

Важной командой этого меню является *Simulation Parameters*. По данной команде открывается диалоговое окно настроек параметров моделирования. Для нас представляют интерес вкладки:

- *Solver* (Расчет) – установка параметров расчета модели;
- *Workspace I/O* (Ввод/вывод данных в рабочую область) – установка параметров обмена данными с рабочей областью MATLAB;
- *Diagnostics* (Диагностика) – выбор уровня диагностики.

Установка параметров управления модельным временем с помощью вкладки *Solver* рис. 5.5.

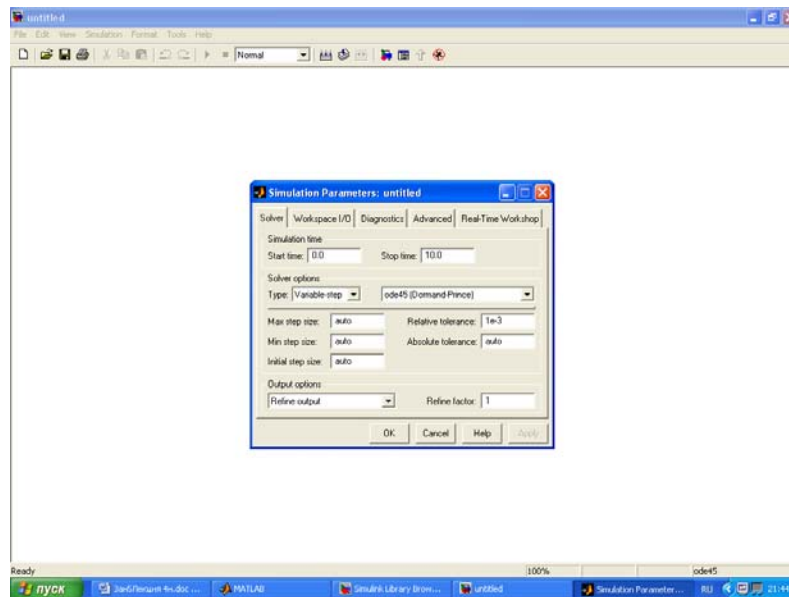


Рис. 5.5. Вкладка *Solver* окна установки параметров моделирования

Параметры моделирования разделены на три группы:

- *Simulation time* (интервал моделирования); величина интервала моделирования задается посредством указания начального (*Start time*) и конечного (*Stop time*) значений модельного времени;
- *Solver options* (параметры расчета) – выбор метода реализации (расчета) модели;
- *Output options* (Параметры вывода) – соответствующие элементы позволяют указать периодичность записи параметров модели в рабочую область (при моделировании с переменным шагом).

Под выбором метода расчета модели понимается следующее. Имея структуру исследуемой системы в виде блок-диаграммы, разработчик может в ходе моделирования выбирать метод отображения динамики системы. С помощью двух раскрывающихся списков *Type* (Тип) система может быть отнесена к одному из следующих классов:

- С дискретным состоянием и дискретным временем перехода из одного состояния в другое.
- С дискретным состоянием и непрерывным временем переходов.
- С непрерывным состоянием и дискретным временем переходов.
- С непрерывным состоянием и непрерывным временем переходов.

Первый список (слева) позволяет выбрать способ изменения модельного времени; он содержит два пункта:

- *Variable-step* (моделирование с переменным шагом);
- *Fixed-step* (моделирование с фиксированным шагом).

Как правило, *Variable-step* используется для моделирования непрерывных систем, а *Fixed-step* – дискретных.

Второй список (справа) позволяет выбрать метод расчета нового состояния системы. Первый вариант (*discrete*) обеспечивает расчет дискретных состояний системы (и для непрерывного, и для дискретного времени переходов из состояния в состояние).

Остальные пункты списка обеспечивают выбор метода расчета нового состояния для непрерывных систем. Эти методы различны для переменного (*Variable-step*) и фиксированного (*Fixed-step*) шага времени, но основаны на единой методике – решении обыкновенных дифференциальных уравнений (ODE).

Ниже двух раскрывающихся списков Type находится область, содержимое которой меняется в зависимости от выбранного способа изменения модельного времени. При выборе **Fixed-step** в данной области появляется текстовое поле Fixed-step size (величина фиксированного шага) позволяющее указывать величину шага моделирования (рис. 5.6). Величина шага моделирования по умолчанию устанавливается системой автоматически (auto). Требуемая величина шага может быть введена вместо значения auto либо в форме числа, либо в виде вычисляемого выражения (то же самое относится и ко всем параметрам, устанавливаемым системой автоматически).

При выборе Fixed-step необходимо также задать режим расчета (Mode). Для параметра Mode доступны три варианта:

- MultiTasking (Многозадачный) – необходимо использовать, если в модели присутствуют параллельно работающие подсистемы и результат работы модели зависит от временных параметров этих подсистем. Режим позволяет выявить несоответствие скорости и дискретности сигналов, пересылаемых блоками друг другу.
- SingleTasking (Однозадачный) – используется для тех моделей, в которых недостаточно строгая синхронизация работы отдельных составляющих не влияет на конечный результат моделирования.

Auto (Автоматический выбор режима) – позволяет Simulink автоматически устанавливать режим MultiTasking для тех моделей, в которых используются блоки с различными скоростями передачи сигналов, и режим SingleTasking для моделей, в которых содержатся блоки, оперирующие одинаковыми скоростями.

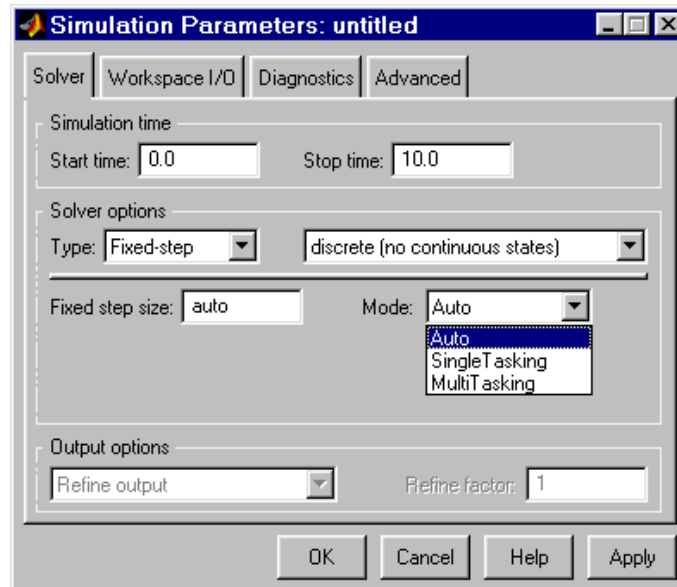


Рис. 5.6. Вкладка Solver при выборе фиксированного шага расчета

При выборе Variable-step в области появляются поля для установки трех параметров:

- Max step size – максимальный шаг расчета. По умолчанию он устанавливается автоматически (auto) и его значение в этом случае равно $(SfopTime - StartTime)/50$. Довольно часто это значение оказывается слишком большим, и наблюдаемые графики представляют собой ломаные (а не плавные) линии. В этом случае величину максимального шага расчета необходимо задавать явным образом.
- Min step size – минимальный шаг расчета.
- Initial step size – начальное значение шага моделирования.

При моделировании непрерывных систем с использованием переменного шага необходимо указать точность вычислений: относительную (Relative tolerance) и абсолютную (Absolute tolerance). По умолчанию они равны соответственно 10^{-3} и auto.

Приведем пример моделирования потока заявок на обслуживание. Предположим, необходимо моделировать поток посетителей супермаркета, подходящих к кассе для оплаты покупки. Наблюдения показали (или из других источников поступила информация), что в вечернее время с 17.00 до 18.00 в среднем через каждые 15 мин. приходит очередной посетитель. Вопрос: как будет выглядеть очередность посетителей на протяжении данного промежутка времени? Модель потока посетителей может выглядеть следующим образом (рис. 5.7).

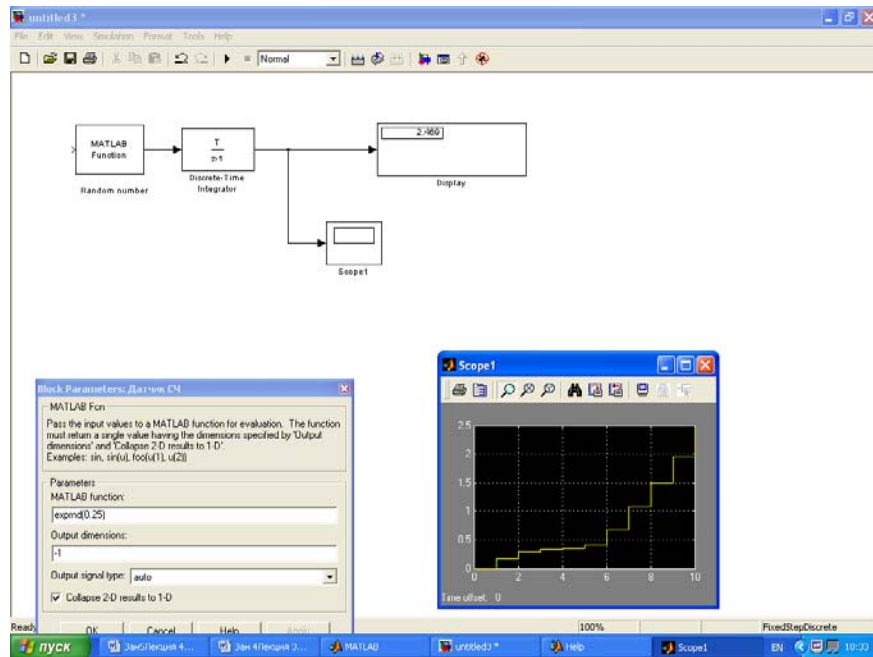


Рис. 5.7. Модель потока посетителей

Блок MATLAB function имитирует случайные числа в соответствии с экспоненциальным законом распределения. С этой целью используется функция *exprnd* с параметром 0.25, что соответствует среднему времени 15 мин. (15/60). Блок Discrete Time Integrator суммирует случайные числа, т.е. формирует время работы кассы нарастающим итогом. Блок Display показывает суммарное время работы кассы (в нашем случае оно будет случайное). Блок Scope1 отражает: по вертикали – общее время работы, которое складывается из случайных временных интервалов; по горизонтали – количество посетителей. С этой осью совпадает шаг моделирования.

В случае моделирования потока покупателей и стоимости их покупки управление модельным временем осуществляется по особым состояниям.

Используя средства Simulink доработаем модель так, чтобы управлять величиной шага моделирования при изменении модельного времени по особым состояниям (рис. 5.8).

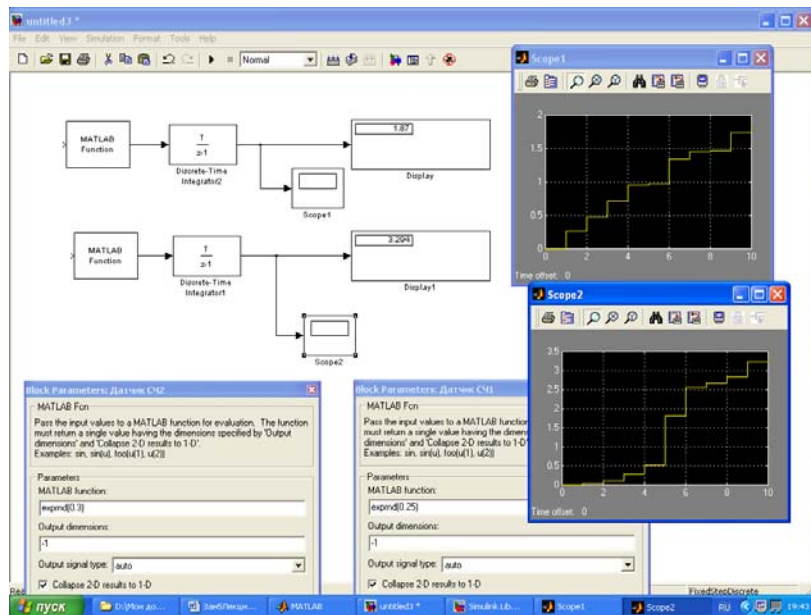


Рис. 5.8. Изменение модельного времени по особым состояниям

Нижняя часть модели обеспечивает формирование отрезков времени, длина которых распределена по заданному закону (экспоненциальному), и продвижение модельного времени. Верхняя часть модели имитирует случайную величину стоимости их покупки. Средняя стоимость покупки в модели принята за 0,3 тыс. руб. Из результатов одного прогона модели видно, что за 3 часа 20 минут в кассе супермаркета будет 1 тыс. 700 рублей. Понятно, что и моделируемая ситуация и сама модель очень просты. Но в данном случае для нас важен подход к решению подобных задач.

Синхронизация параллельных процессов

До этого мы рассматривали только асинхронные параллельные процессы, т.е. такие, которые не влияют друг на друга. «Привязку» таких процессов к единой оси модельного времени Simulink выполняет сам, освобождая от соответствующих проблем разработчика. Другое дело, когда имеют место синхронные параллельные процессы, состояние каждого из которых зависит от текущего состояния другого. При согласовании таких процессов должна учитываться специфика решаемой задачи, и без помощи разработчика Simulink здесь уже обойтись не может.

Для корректного управления модельным временем в таких моделях необходимо:

1. Установить, какой из взаимодействующих процессов является подчиненным по отношению к другому.
2. Определить, могут ли в подчиненном процессе происходить события, не связанные с изменением состояния управляющего процесса.
3. Обеспечить приращение модельного времени на интервал времени до ближайшего события в управляющем процессе.
4. Контролировать условия окончания сеанса моделирования.

Пример. В качестве иллюстрации к сказанному вновь воспользуемся моделью работы супермаркета. Очевидно, что оплата покупателя в кассу может начаться только при условии под-

хода его к кассе. Потому процесс оплаты является подчиненным по отношению к процессу подхода покупателей к кассе. Если интервал между покупателями значительно больше длительности расчета в кассе, то продвижение модельного времени будет определяться только событиями управляющего процесса (поток покупателей). Блок-диаграмма такой системы не будет отличаться от рассмотренной ранее (см. рис. 5.8). Если же интервал между покупателями в кассе соизмерим с временем обслуживания в кассе, то необходимо отразить в модели дополнительное условие: «обслуживание» очередного покупателя не может начаться до тех пор, пока не завершится «обслуживание» предыдущего. При такой постановке задачи оба процесса становятся равноправными с точки зрения влияния на значение модельного времени. Очередной шаг модельного времени в этом случае вычисляется как сумма двух временных интервалов: промежутка до нового покупателя и длительности обслуживания его в кассе. Модель будет иметь вид, показанный на рис. 5.9. Результаты работы модели представлены на рис. 5.10–5.13.

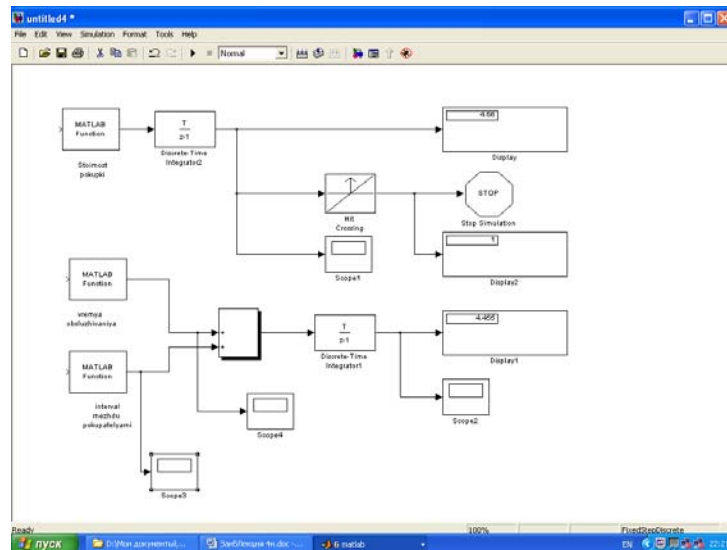


Рис. 5.9. Управление модельным временем при наличии двух синхронных процессов

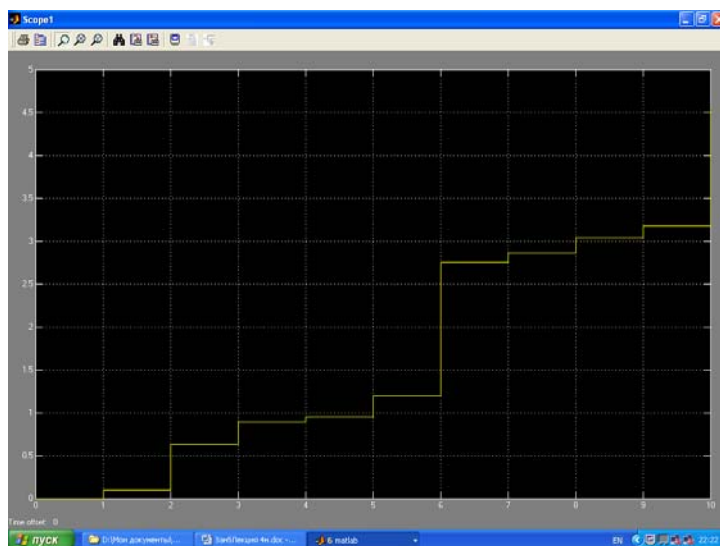


Рис. 5.10. Результат работы модели (поступление денег в кассу)

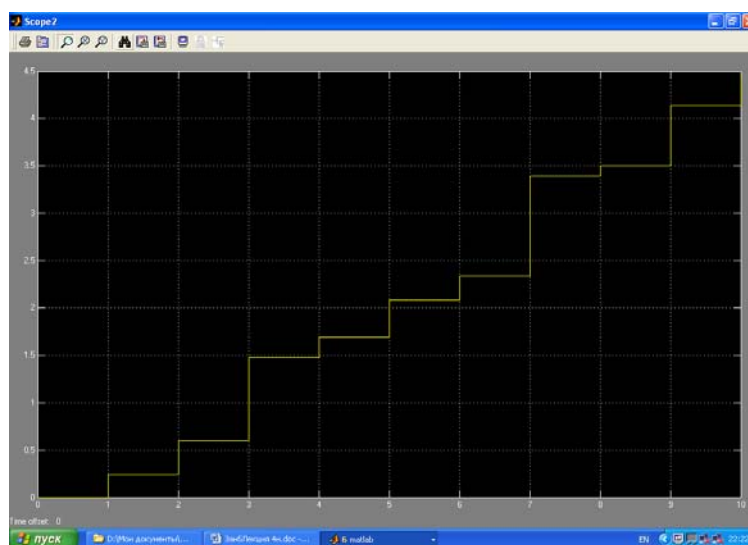


Рис. 5.11. Результаты работы модели (суммарный интервал между покупателями и время обслуживания покупателя – по вертикали)

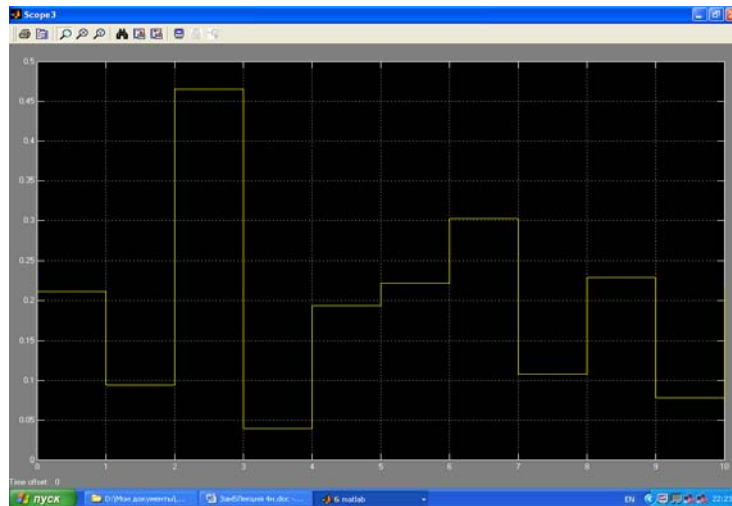


Рис. 5.12. Результат работы модели (интервал между покупателями - по вертикали)

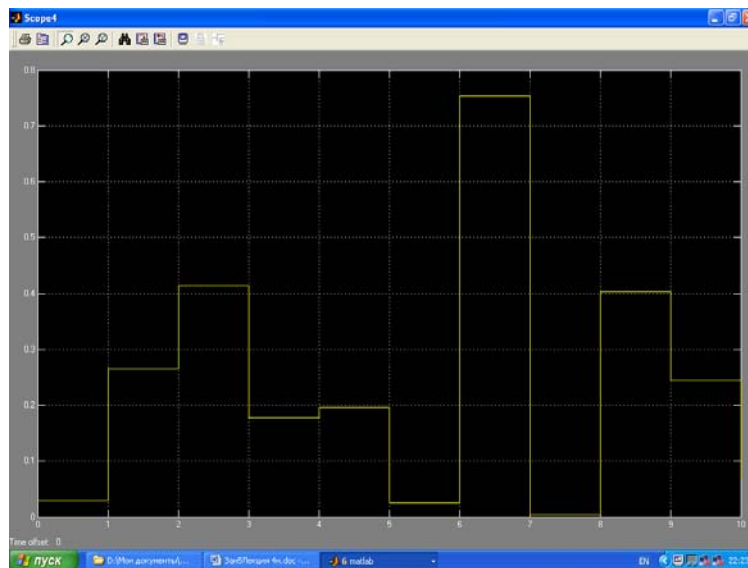


Рис. 5.13. Результат работы модели (время обслуживания покупателя - по вертикали)

При моделировании аналогичного процесса, но с двумя кассами, необходимо учитывать, что они работают одновременно и совершенно независимо друг от друга, т.е. асинхронно. Предположим, что сеанс моделирования должен быть остановлен по истечении 300 единиц модельного времени. Модель будет иметь вид, представленный на рис. 5.14.

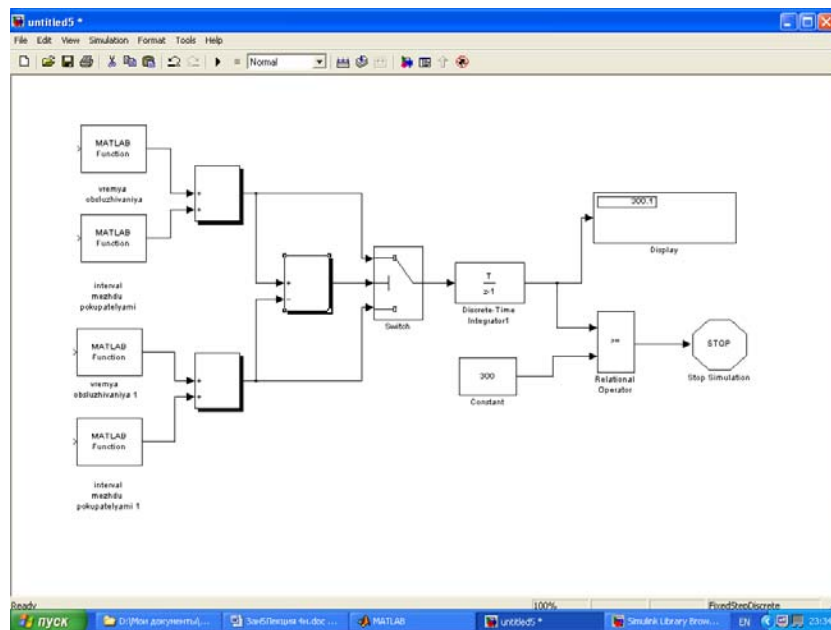


Рис. 5.14. Управление окончанием моделирования при наличии двух параллельных процессов

В данной модели используется способ продвижения модельного времени, суть которого состоит в том, что в каждом такте выбирается большая из величин T_1 и T_2 . Где T_1 и T_2 – время суммарного обслуживания очередного покупателя в 1-й и во 2-й кассе соответственно.

При этом исходим из того, что суммарное время обслуживания каждого покупателя состоит из двух составляющих:

- Ожидание (в течение времени $t_{ож}$);
- Обслуживание в кассе ($t_{обсл}$).

$$T1 = t_{ож1} + t_{обсл1}$$

$$T2 = t_{ож2} + t_{обсл2}$$

§6. Установка параметров вывода выходных сигналов моделируемой системы OUTPUT OPTIONS (параметры вывода)

В нижней части вкладки Solver задаются настройки параметров вывода выходных сигналов моделируемой системы (Output options). Для данного параметра возможен выбор одного из трех вариантов:

- **Refine output** (Скорректированный вывод) – позволяет изменять дискретность регистрации модельного времени и тех сигналов, которые сохраняются в рабочей области MATLAB с помощью блока To Workspace. Установка величины дискретности выполняется в строке редактирования Refine factor, расположенной справа. По умолчанию значение Refine factor равно 1, это означает, что регистрация производится с шагом $Dt = 1$ (то есть для каждого значения модельного времени). Если задать Refine factor равным 2, это означает, что будет регистрироваться каждое второе значение сигналов, 3 – каждое третье и т.д. Параметр Refine factor может принимать только целые положительные значения.
- **Produce additional output** (Дополнительный вывод) – обеспечивает дополнительную регистрацию параметров модели в заданные моменты времени; их значения вводятся в строке редактирования (в этом случае она называется Output times) в виде списка, заключенного в квадратные скобки. При использовании этого варианта базовый шаг регистрации (Dt) равен 1.

Значения времени в списке Output times могут быть дробными числами и иметь любую точность.

- **Produce specified output only** (Формировать только заданный вывод) – устанавливает вывод параметров модели только в заданные моменты времени, которые указываются в поле Output times (Моменты времени вывода).

§7. Установка параметров обмена с рабочей областью

Элементы, позволяющие управлять вводом и выводом в рабочую область MATLAB промежуточных данных и результатов моделирования, расположены на вкладке Workspace I/O (рис. 5.15).

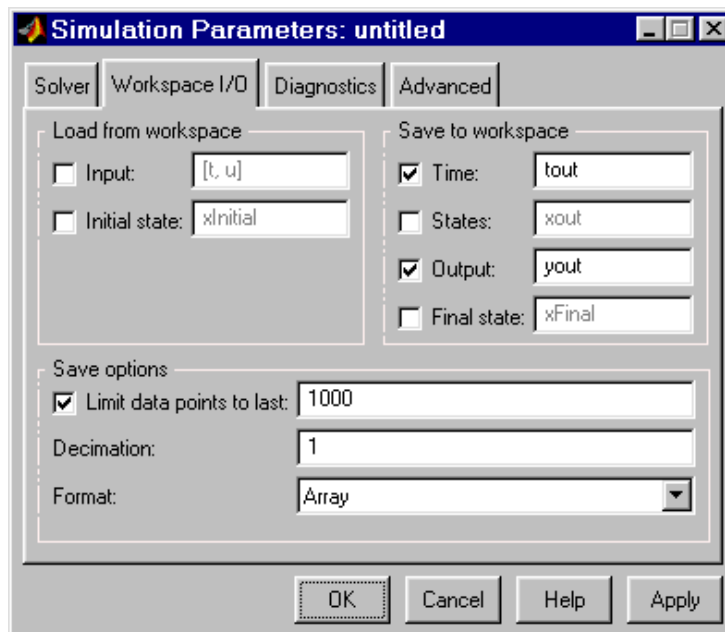


Рис. 5.15. Вкладка Workspace I/O диалогового окна установки параметров моделирования

Элементы вкладки разделены на 3 поля:

- **Load from workspace** (Загрузить из рабочей области). Если флажок Input (Входные данные) установлен, то в расположенном справа текстовом поле можно ввести формат данных, которые будут считываться из рабочей области MATLAB. Установка флажка Initial State (Начальное состояние) позволяет ввести в связанном с ним текстовом поле имя переменной, содержащей параметры начального состояния модели. Данные, указанные в полях Input и Initial State, передаются в исполняемую модель посредством одного или более блоков In (из раздела библиотеки Sources).
- **Save to workspace** (Записать в рабочую область) – позволяет установить режим вывода значений сигналов в рабочую область MATLAB и задать их имена.
- **Save options** (Параметры записи) – задает количество строк при передаче переменных в рабочую область. Если флажок Limit rows to last установлен, то в поле ввода можно указать количество передаваемых строк (отсчет строк производится от момента завершения расчета). Если флажок не установлен, то передаются все данные. Параметр Decimation (Исключение) задает шаг записи переменных в рабочую область (аналогично параметру Refine factor вкладки Solver). Параметр Format (формат данных) задает формат передаваемых в рабочую область данных. Доступные форматы Array (Массив), Structure (Структура), Structure With Time (Структура с дополнительным полем – «время»).

§8. Установка параметров диагностирования модели

Вкладка Diagnostics (рис. 5.16) позволяет изменять перечень диагностических сообщений, выводимых Simulink в командном окне MATLAB, а также устанавливать дополнительные параметры диагностики модели.

Сообщения об ошибках или проблемных ситуациях, обнаруженных Simulink в ходе моделирования и требующих вмешательства разработчика, выводятся в командном окне MATLAB. Исходный перечень таких ситуаций и вид реакции на них приведен в списке на вкладке Diagnostics. Разработчик может указать вид реакции на каждое из них, используя группу переключателей в поле Action (они становятся доступны, если в списке выбрано одно из событий):

- **None** – игнорировать;
- **Warning** – выдать предупреждение и продолжить моделирование;
- **Error** – выдать сообщение об ошибке и остановить сеанс моделирования.

Выборный вид реакции отображается в списке рядом с наименованием события.

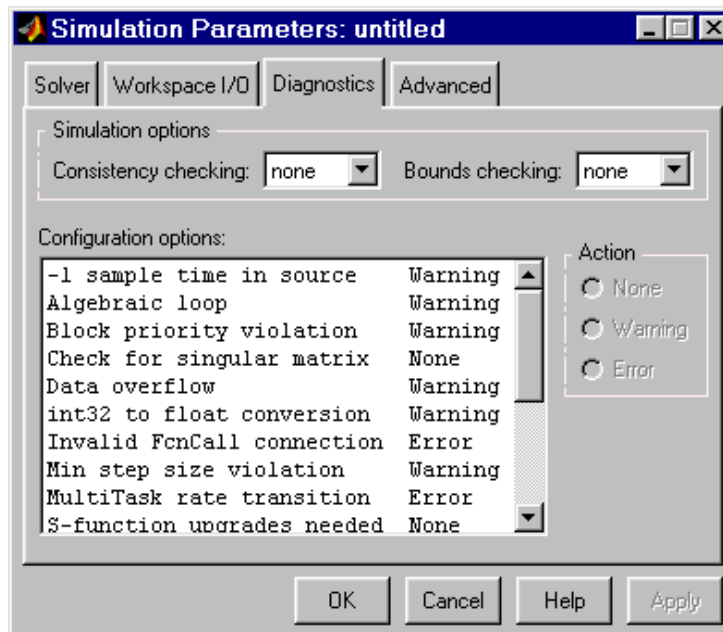


Рис. 5.16. Вкладка Diagnostics окна установки параметров моделирования

Раздел III.

Основные правила моделирования

Глава 6. Классификация математических моделей экономических систем

В курсе «Имитационное моделирование экономических процессов» рассматриваются не сами экономические системы, а лишь методы разработки имитационных моделей этих систем. Разработкой классификации экономических систем должны заниматься экономисты. Насколько можно судить по литературным источникам общепринятой классификации, таких систем пока не существует.

Первая классификация математических моделей экономических систем была приведена в монографии Т. Нейлора «Машинные имитационные эксперименты с моделями экономических систем» в 1971 г. Он предлагал разделить их все на две группы (рис. 6.1):

- 1) общие экономические модели;
- 2) модели управления предприятиями.

§1. Общие экономические модели

В основу классификации общих экономических моделей заложен масштаб изучаемой экономической системы. С этой точки зрения модели можно разделить на три большие группы: модели фирм, отраслевые модели и макроэкономические модели.



Рис. 6.1. Классификация алгоритмических экономических моделей по Т. Нейлору

Модели фирм

Разновидности моделей фирм:

- модели отдельных фирм;
- модели конкурентных отраслей;
- модели дуополий (объединений двух фирм);
- модели олигополии (объединений нескольких фирм);
- модели монополий.

Опыт создания моделей фирм в США, обобщенный Т. Нейлором, показывает, что разработка математических моделей даже для систем такого масштаба, как фирма, представляет сложную научно-исследовательскую проблему.

Во-первых, это проблема получения достоверной информации. Модель должна строиться на прочной эмпирической основе. Однако эта информация, как правило, недоступна для разработчиков экономических моделей. Руководство компаний просто не желает давать данные о деятельности своих предприятий посторонним лицам. Это особенно характерно для фирм, работающих в условиях сильной конкуренции.

Во-вторых, трудности построения адекватной численной модели фирмы связаны с тем, что такая модель должна опираться на глубокое знание реальных процессов принятия решений в организациях. Для этого надо хорошо ориентироваться в современном состоянии таких дисциплин, как теория принятия решений, теория организации, а также разбираться в вопросах психологии, социологии, политики, управления производством и экономики.

В-третьих, организация численных испытаний модели функционирования фирмы требует особого внимания к проблеме планирования эксперимента,

В итоге применение мощного аппарата имитационного моделирования оказывается неэффективным, так как в этих условиях традиционные аналитические методы дают не менее надежные результаты.

В качестве примера в 9 главе рассматривается одна из моделей фирмы, получившая название «паутинообразной» модели. Это простейшая динамическая модель взаимодействия фирмы и рынка. В качестве типовой математической схемы вначале была выбрана непрерывно-детерминированная модель, в которую добавлены случайные составляющие входных переменных, в результате чего модель становится стохастической. Особенностью рассматриваемого варианта модели также является то, что это модель с обучением, т.е. с учетом тенденции развития моделируемого процесса.

К **отраслевым моделям** относятся комплексные, или агрегированные, модели, описывающие отдельные отрасли народного хозяйства как единое целое. В монографии Т. Нейлора дается краткая характеристика модели текстильной

промышленности США, модели кожевенной и обувной промышленности, а также модели лесобработывающей промышленности. В большинстве случаев отраслевая модель представляет собой систему рекуррентных уравнений. Для нахождения коэффициентов этих уравнений используется метод наименьших квадратов.

Макроэкономические модели предназначены для имитации экономических систем крупного масштаба, таких как область или страна в целом. С чисто технической точки зрения механизм численного моделирования экономики описывается теми же правилами, что и при имитации процесса функционирования фирмы, отрасли или их подразделений. Здесь по-прежнему необходимо определить структуру изучаемой системы, входные и выходные переменные, сформулировать задачу моделирования, построить схему модели и реализовать ее, например в Matlab. Однако по существу имитационные модели глобальных экономических систем сильно отличаются от микроэкономических моделей, и эти отличия связаны с проблемой вывода адекватных уравнений функционирования экономики в целом:

- входные переменные макроэкономической системы, такие как национальный доход, национальный продукт и общая численность работающих, по-видимому, зависят от большого числа существенных факторов. Их количество обычно намного превышает число переменных, рассматриваемых в численных микроэкономических моделях;
- возникает проблема «агрегирования» микроэкономических переменных в обобщенные показатели макроэкономической системы;
- между входными переменными существуют очень сложные взаимодействия и обратные связи;
- для формулировки реалистических гипотез относительно функционирования экономики требуются глубокие знания закономерностей ее развития. Ис-

следователь, пытающийся восполнить недостаток этих знаний с помощью имитационных экспериментов, скорее построит модель собственного невежества, а не реального мира;

- получить данные для построения математической модели макроэкономической системы намного сложнее, чем для микроэкономических систем.

По приведенным причинам в данной дисциплине отраслевые и макроэкономические модели не рассматриваются.

§2. Модели управления предприятием

Это микроэкономические модели, отличающиеся друг от друга не столько областью применения, сколько тем, какая типовая математическая схема заложена в основу модели и каковы особенности используемого математического аппарата.

К моделям управления предприятиями относятся (см. рис. 6.1):

- модели массового обслуживания;
- модели управления запасами;
- производственные модели;
- модели торговли;
- финансовые модели.

Для многих промышленных систем характерен поток входных требований (заявок), поступающих в один или несколько каналов обслуживания и иногда образующих очередь. Заявками могут быть производственные и торговые заказы, заявки на ремонт станков, посадку самолетов в аэропорту и заправку автомобилей на автозаправочной станции. Канал обслуживания может представлять собой совокупность устройств, этап производственного процесса, аэропорт или театральную кассу. Интервалы между последовательными заявками и продолжительность их обслуживания являются случайными величинами.

К **моделям системы массового обслуживания** относятся модель бензоколонки, нотариальной конторы, парик-

махерской, столовой самообслуживания, станции автомобильного обслуживания и т.п. Общей для этих моделей является заложенная типовая математическая схема – схема системы массового обслуживания с переменным числом каналов, однородным потоком заявок, без отказов и с ограниченным ожиданием. В задачу моделирования входит установление оптимального числа каналов, которое при определенном соотношении входных параметров (среднего времени между соседними заявками и среднего времени обслуживания) обеспечивает максимальное значение показателя эффективности процесса функционирования системы. Для конкретной экономической системы в качестве критерия эффективности используется условие максимума прибыли.

К **производственным моделям** относится имитационная модель производственной фирмы, включающей несколько цехов, которые последовательно участвуют в процессе производства некоторого изделия. Заказы на изготовление изделия поступают нерегулярно (в случайные моменты времени). При оптимальной структуре предприятия (количестве цехов) и оптимальном распределении производственных ресурсов обеспечивается максимум прибыли. Имитация работы предприятия производится с помощью модели одноканальной многофазной системы массового обслуживания, без отказов с неограниченным ожиданием.

К моделям этого же класса относятся модели, условно названные моделями управленческого звена учреждения, фирмы или предприятия, состоящего из начальника (заведующего) и двух его заместителей. Все они принимают участие в процессе приема посетителей или обработки документации. Часть посетителей, побывавших на приеме у одного из заместителей, затем отправляются на прием к начальнику. При определенном соотношении параметров системы можно обеспечить практически одинаковую занятость каждого из трех должностных лиц. С точки зрения используемого математического аппарата это модель двухканальной двухфазной системы массового обслуживания с двумя приоритетами заявок, без отказов с неограниченным ожиданием.

Обширную группу промышленных систем, при изучении которых эффективна численная имитация, образуют так называемые **системы хранения запасов**. Большинство задач управления запасами сводится к поиску оптимального распределения поставок в моделируемую систему. Модель должна дать ответ на вопрос: сколько следует фирме заказывать (или производить) и как часто она должна повторять заказы, чтобы минимизировать сумму издержек хранения запаса, издержек, связанных с организацией поставок, и потерь вследствие недостатка продукта на складе?

К **моделям управления запасами** относятся модели системы управления запасами однородного товара на складе. Предполагается, что, когда уровень запаса падает ниже некоторой критической отметки, оформляется заказ на поставку новой партии товара. При отсутствии товара на складе применяются штрафные санкции. При определенном соотношении параметров системы суммарные расходы на содержание склада могут быть минимизированы.

С точки зрения используемого математического аппарата это имитационная модель, в которой две входные переменные (дневной спрос и время выполнения заказа) являются случайными величинами, что определяет случайный характер выходной характеристики – суммарных издержек, характеризующих работу склада за определенный период времени. Время между соседними заявками на приобретение товара и время на выдачу товара в модели не фигурируют. Поэтому эту модель нельзя отнести к классу моделей СМО, однако это тоже непрерывно-стохастическая модель.

К **группе моделей торговли** относится так называемая модель фирмы. Примером такой модели может быть модель выездной торговой точки, которая может вести торговлю в различных пунктах с различными условиями при действии случайных факторов. Задача состоит в установлении закономерностей моделируемого процесса и условий, при которых пункты торговли могут считаться эквивалентными по получаемой прибыли.

В качестве типовой математической схемы здесь использована общая непрерывно-стохастическая модель, в которой имитируется влияние дискретных и непрерывных случайных факторов. Такая модель при изменении комплекта исходных данных может использоваться как вариант транспортной модели, в которой осуществляется имитация процесса перевозки грузов по нескольким маршрутам в условиях влияния случайных помех и непостоянства скорости движения на различных участках дороги.

С помощью **финансовой модели (модель инвестиций)** определяется объем капиталовложений в условиях неопределенности. Примером такой модели является модель инвестиционной компании, предполагающей вложить свои средства в строительство нового предприятия. Предприятие будет выпускать продукцию, пользующуюся спросом на рынке. Модель должна оценивать минимальную гарантированную прибыль от продажи продукции в условиях конкуренции на рынке. Финансовая модель позволяет оценивать риски инвесторов. Используемая типовая математическая схема представляет собой непрерывно-стохастическую модель. Для раскрытия неопределенности необходимо выбрать в качестве одной из входных переменных случайную величину, имеющую произвольное дискретно-непрерывное распределение.

Глава 7. Моделирование процессов обслуживания заявок в условиях отказов

В экономических системах могут возникать отказы. Появление их обуславливается отказами в технических подсистемах, отсутствием временных или материальных ресурсов и т.д. Различают два рода отказов.

Отказы первого рода (неисправности) приводят к временному прекращению процесса обслуживания очередной заявки с сохранением достигнутого состояния. После устранения отказа процесс обслуживания заявки может продолжаться. В качестве примера можно привести отказ оборудования бензоколонки. После устранения неисправности заправка автомашины продолжается.

Отказы второго рода (аварии) приводят к такому состоянию системы, когда после устранения отказа процесс обслуживания заявки начинается сначала. Примером может служить временное отключение электропитания при работе персонального компьютера во время решения расчетной задачи. После устранения аварии процесс решения задачи начинается сначала.

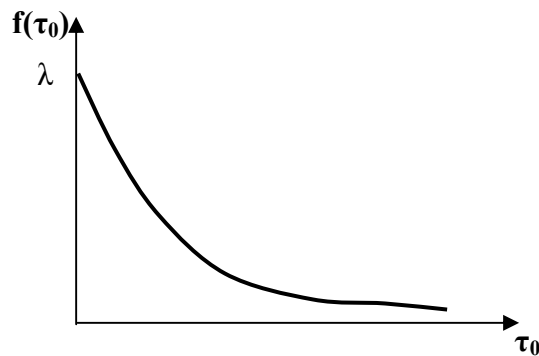
Время возникновения отказов в системе следует считать случайным событием. Период устранения отказа также может рассматриваться как случайный отрезок времени. Принято считать, что период безотказной работы и период устранения отказа имеют показательные распределения с определенными параметрами.

Функция плотности для времени безотказной работы

$$f(\tau_0) = \lambda_0 \exp(-\lambda_0 \tau_0),$$

где τ_0 – время безотказной работы;

λ_0 – параметр (интенсивность потока отказов, т.е. количество отказов в единицу времени).



Функция плотности для времени устранения отказа

$$f(\tau_y) = \lambda_y \exp(-\lambda_y \tau_y),$$

где τ_y – время устранения отказа;

λ_y – параметр (среднее число устраненных отказов в единицу времени).

Особенностью взаимодействия периодов безотказной работы и периодов устранения отказов является то, что они не могут пересекаться или накладываться друг на друга. Эти периоды должны чередоваться. Поэтому интервал между двумя соседними отказами должен рассматриваться как сумма (композиция) двух распределений случайных величин τ_y и τ_0 .

Можно показать, что композиция этих распределений приводит к обобщенному *потoku Эрланга 2-го порядка*, плотность которого имеет вид:

$$f(\tau) = \frac{\lambda_0 \lambda_y [\exp(-\lambda_0 \tau) - \exp(-\lambda_y \tau)]}{\lambda_y - \lambda_0}.$$

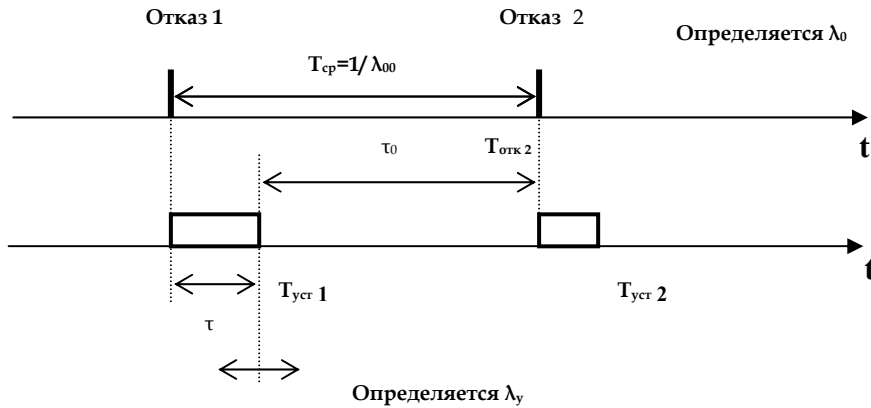


Рис. 7.1. Временные диаграммы потока отказов и их устранения

Момент времени появления очередного отказа определяется

$$T_{\text{отк } i} = T_{\text{ср}} * \eta - T_{\text{уст } i-1},$$

где $T_{\text{уст } i-1} = T_{\text{уст. ср}} * \eta - T_{\text{отк } i-1}$.

Данные выражения берутся в основу моделирования отказов. Блок модели формирования одиночного отказа может быть выполнен, например, по схеме, представленной на рис. 7.2.

$$T_{\text{отк } i} = \frac{T_{\text{ср}} * \text{exprnd}(\lambda_0) - (T_{\text{уст. ср}} * \text{exprnd}(\lambda_0) + T_{\text{отк } i-1})}{F_{\text{сп}}}$$

Рис. 7.2. Блок модели формирования одиночного отказа

Формирование потока отказов осуществляется многократным запуском блока формирования отказа. Программа Matlab позволяет решать эту задачу иным способом.

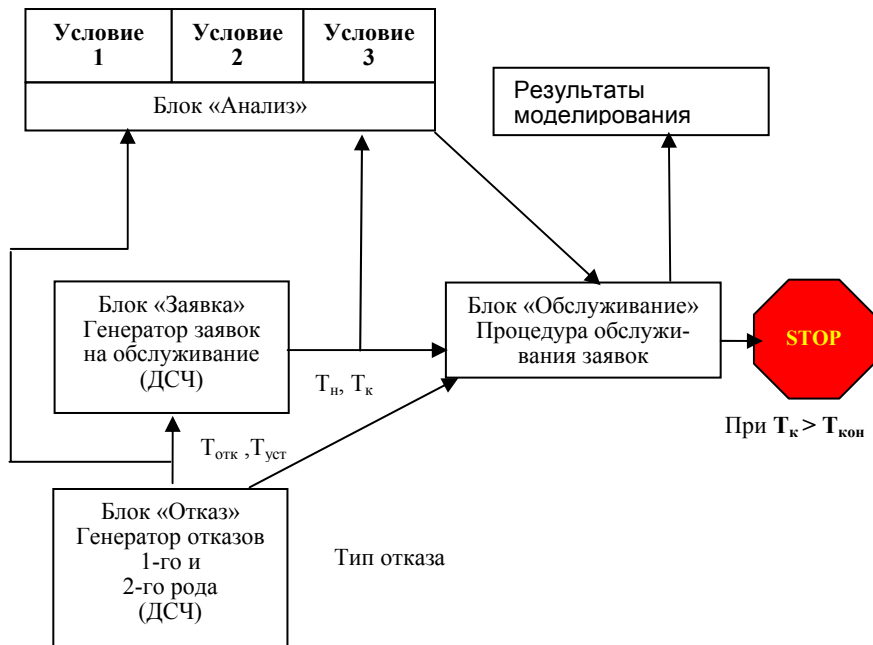


Рис. 7.3. Блок-схема имитационной модели одноканальной системы массового обслуживания (СМО) с отказами

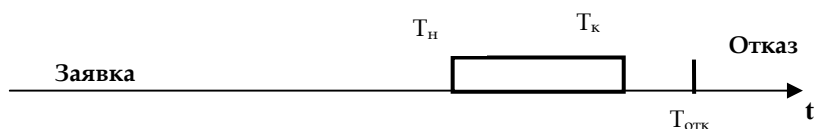
Блок «Заявка». Генератор заявок формирует массив случайных чисел времен появления заявок на обслуживание с учетом возможного времени ожидания обслуживания и без учета возможности отказа (T_n). Здесь же определяется возможное время окончания обслуживания без учета возможности появления отказа (T_k). В блоке используется ДСЧ с показательным законом распределения.

Блок «Отказ». Блок формирует последовательности случайных чисел $T_{отк}$ и $T_{уст}$, а также с помощью ДСЧ формирует тип отказа.

Блок «Анализ».

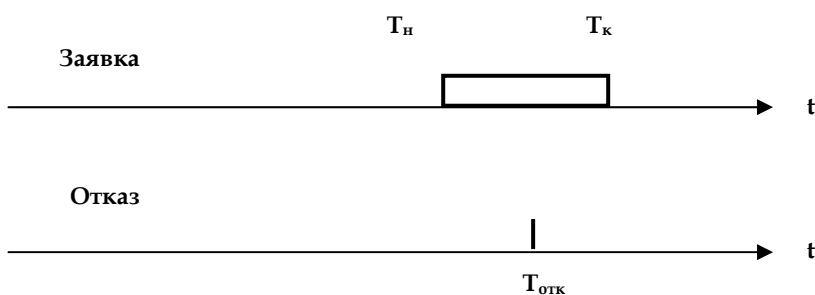
Блок осуществляет проверку следующих условий.

Условие 1. $T_k < T_{отк}$



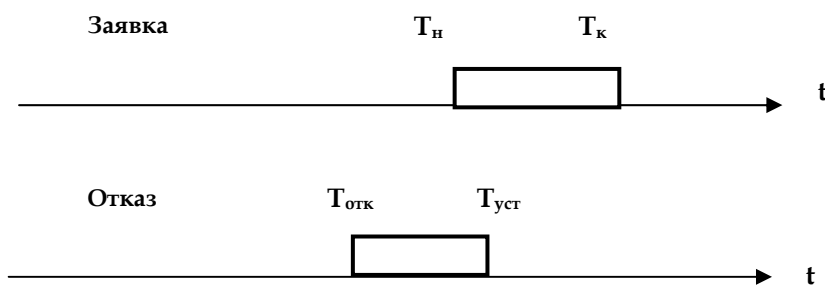
Отказ появился после того, как процесс обслуживания заявки был полностью завершен.

Условие 2. $((T_n < T_{отк}) \text{ И } (T_{отк} < T_k))$



Обслуживание заявки будет прекращено из-за отказа и продолжено после устранения (отказ 1-го рода) или обслуживание начнется заново (отказ 2-го рода).

Условие 3. $((T_{отк} < T_n) \text{ И } (T_n < T_{уст}))$



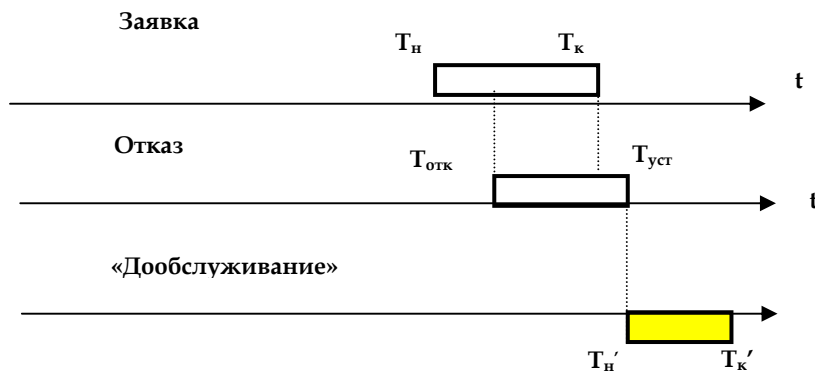
Заявка поступила в момент, когда происходит устранение отказа.

Блок «Обслуживание». Блок в зависимости от выполняемого условия в блоке «Анализ» производит:

1. Подсчет количества обслуженных заявок (если выполняется условие 1).
2. Корректировку времени начала и окончания «дообслуживания» (если выполняется условие 2) по формулам

$$T'_H = T_{уст},$$

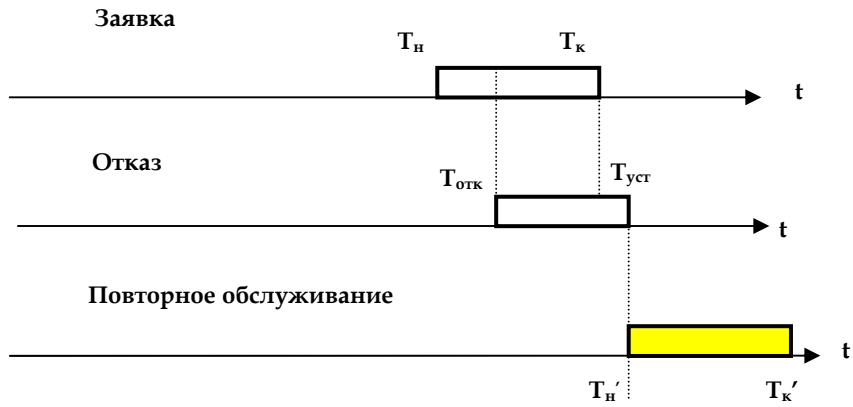
$$T'_K = T_K + T_{уст} - T_{отк}.$$



3. Корректировку времени (если выполняется условие 3, т.е. выполнение заявки прервал отказ второго рода или заявка поступила в момент, когда происходит устранение отказа) по формулам

$$T'_H = T_{уст}$$

$$T'_K = T_K + T_{уст} - T_H.$$



Работа модели происходит последовательно, т.е. после обслуживания очередной заявки моделируется обслуживание следующей.

Моделирование заканчивается в случае, если закончится период функционирования системы, т.е. выполнится условие

$$T_K > T_{кон}.$$

Глава 8. Планирование модельных экспериментов

§1. Цели планирования экспериментов

Для правильной организации модельного эксперимента исследователь должен располагать следующей информацией:

- 1) к какому классу относится моделируемая система (статическая или динамическая, детерминированная или стохастическая и т.д.);
- 2) какой режим работы системы его интересует: стационарный (установившийся) или нестационарный;
- 3) в течение какого промежутка времени следует наблюдать за поведением (функционированием) системы;
- 4) какой объем испытаний (т.е. повторных экспериментов) сможет обеспечить требуемую точность оценок (в статистическом смысле) исследуемых характеристик системы.

Разумеется, можно пойти по такому пути: не особенно задумываясь над перечисленными вопросами, взять от модели все «по максимуму» – исследовать работу системы во всех режимах, для всех возможных сочетаний внешних и внутренних параметров и повторять каждый эксперимент по сотне раз. Однако польза от такого моделирования невелика, поскольку полученные данные будет очень сложно обработать и проанализировать, а еще труднее принять с их помощью какое-либо конкретное решение. Да и затраты времени на моделирование, даже с учетом быстродействия современных компьютеров, окажутся чрезмерными.

Таким образом, планирование модельных экспериментов преследует две основные цели:

- сокращение общего объема испытаний при соблюдении требований к достоверности и точности их результатов;
- повышение информативности каждого из экспериментов в отдельности.

Поиск плана эксперимента производится в так называемом факторном пространстве.

Факторное пространство – это множество внешних и внутренних параметров модели, значения которых исследователь может контролировать в ходе подготовки и проведения модельного эксперимента.

Во многих случаях факторы могут носить не только количественный, но и качественный характер. Поэтому значения факторов обычно называют уровнями. Если при проведении эксперимента исследователь может изменять уровни факторов, эксперимент называется активным, в противном случае – пассивным.

Введем еще несколько терминов, используемых в теории планирования эксперимента. Каждый из факторов имеет верхний и нижний уровни, расположенные симметрично относительно некоторого нулевого уровня. Точка в факторном пространстве, соответствующая нулевым уровням всех факторов, называется *центром плана*.

Интервалом варьирования фактора называется некоторое число, прибавление которого к нулевому уровню дает верхний уровень, а вычитание – нижний. Как правило, план эксперимента строится относительно одного (основного) выходного скалярного параметра Y , который называется *наблюдаемой переменной*. Если моделирование используется как инструмент принятия решения, то в роли наблюдаемой переменной выступает показатель эффективности. При этом предполагается, что значение наблюдаемой переменной, полученное в ходе эксперимента, складывается из двух составляющих

$$Y = f(x) + e(x),$$

где $f(x)$ – функция отклика (неслучайная функция факторов);

$e(x)$ – ошибка эксперимента (случайная величина);

x – точка в факторном пространстве (определенное сочетание уровней факторов).

Очевидно, что y является случайной переменной, так как зависит от случайной величины $e(x)$.

Дисперсия Dy наблюдаемой переменной, которая характеризует точность измерений, равна дисперсии ошибки опыта: $Dy = De$.

Dy называют дисперсией воспроизводимости эксперимента. Она характеризует качество эксперимента. Эксперимент называется идеальным при $Dy = 0$.

Существует два основных варианта постановки задачи планирования имитационного эксперимента:

1. Из всех допустимых выбрать такой план, который позволил бы – получить наиболее достоверное значение функции отклика $f(x)$ при фиксированном числе опытов.

2. Выбрать такой допустимый план, при котором статистическая оценка функции отклика может быть получена с заданной точностью при минимальном объеме испытаний.

Решение задачи планирования в первой постановке называется стратегическим планированием эксперимента, во второй – тактическим планированием.

§2. Стратегическое планирование имитационного эксперимента

Итак, цель методов стратегического планирования имитационных экспериментов – получение максимального объема информации об исследуемой системе в каждом эксперименте (наблюдении). Другими словами, стратегическое планирование позволяет ответить на вопрос, при каком сочетании уровней внешних и внутренних факторов может быть получена наиболее полная и достоверная информация о поведении системы.

При стратегическом планировании эксперимента должны быть решены две основные задачи:

1. Идентификация факторов.
2. Выбор уровней факторов.

Под *идентификацией факторов* понимается их ранжирование по степени влияния на значение наблюдаемой переменной (показателя эффективности).

По итогам идентификации целесообразно разделить все факторы на две группы – первичные и вторичные. *Первичные* –

это те факторы, в исследовании влияния которых экспериментатор заинтересован непосредственно. *Вторичные* – факторы, которые не являются предметом исследования, но влиянием которых нельзя пренебречь.

Выбор уровней факторов производится с учетом двух противоречивых требований:

- уровни фактора должны перекрывать (заполнять) весь возможный диапазон его изменения;
- общее количество уровней по всем факторам не должно приводить к чрезмерному объему моделирования. Поиск компромиссного решения, удовлетворяющего этим требованиям, и является задачей стратегического планирования эксперимента.

Способы построения стратегического плана

Эксперимент, в котором реализуются всевозможные сочетания уровней факторов, называется *полным факторным экспериментом* (ПФЭ).

Общее число различных комбинаций уровней в ПФЭ для k -факторов можно вычислить следующим образом:

$$N = l_1 \cdot l_2 \cdot l_3 \dots l_k,$$

где l_k – число уровней k -го фактора.

Если число уровней для всех факторов одинаково, то $N = L^k$ (L – число уровней).

Недостаток ПФЭ – большие временные затраты на подготовку и проведение.

Например, если в модели отражены 3 фактора, влияющие на значение выбранного показателя эффективности, каждый из которых имеет 4 возможных уровня (значения), то план проведения ПФЭ будет включать 64 эксперимента ($N = 4^3$). Если при этом каждый из них длится хотя бы одну минуту (с учетом времени на изменение значений факторов), то на однократную реализацию ПФЭ потребуется более часа.

Поэтому использование ПФЭ целесообразно только в том случае, если в ходе имитационного эксперимента исследуется взаимное влияние всех факторов, фигурирующих в модели.

Если такие взаимодействия считают отсутствующими или их эффектом пренебрегают, проводят *частичный факторный эксперимент* (ЧФЭ).

Известны и применяются на практике различные варианты построения планов ЧФЭ. Мы рассмотрим только некоторые из них.

1. Рандомизированный план – предполагает выбор сочетания уровней для каждого прогона случайным образом. При использовании этого метода отправной точкой в формировании плана является число экспериментов, которые считает возможным (или необходимым) провести исследователь.

2. Латинский план (или «латинский квадрат») – используется в том случае, когда проводится эксперимент с одним первичным фактором и несколькими вторичными. Суть такого планирования состоит в следующем. Если первичный фактор А имеет l уровней, то для каждого вторичного фактора также выбирается l уровней. Выбор комбинации уровней факторов выполняется на основе специальной процедуры, которую мы рассмотрим на примере.

Пусть в эксперименте используется первичный фактор А и два вторичных фактора – В и С, число уровней факторов l равно 4. Соответствующий план можно представить в виде квадратной матрицы размером $l \times l$ (4×4) относительно уровней фактора А. При этом матрица строится таким образом, чтобы в каждой строке и в каждом столбце данный уровень фактора А встречался только один раз.

В результате имеем план, требующий $4 \times 4 = 16$ прогонов, в отличие от ПФЭ, для которого нужно $4^3 = 64$ прогона.

3. Эксперимент с изменением факторов по одному. Суть его состоит в том, что один из факторов «пробегают» все l уровней, а остальные $n-1$ факторов поддерживаются постоянными. Такой план обеспечивает исследование эффектов каждого фактора в отдельности. Он требует всего $N = l_1 + l_2 + l_3 + \dots + l_k$ прогонов.

Для рассмотренного выше примера (3 фактора, имеющие по 4 уровня) $N = 4 + 4 + 4 = 12$.

Еще раз подчеркнем, что такой план применим (как и любой ЧФЭ) только при отсутствии взаимодействия между факторами.

4. Дробный факторный эксперимент. Каждый фактор имеет два уровня – нижний и верхний, поэтому общее число вариантов эксперимента $N = 2^k$, где k – число факторов. Матрицы планов для $k = 2$ и $k = 3$ приведены ниже.

Планы, построенные по такому принципу, обладают определенными свойствами (симметричность, нормированность, ортогональность и ротатабельность), обеспечивающими повышение качества проводимых экспериментов.

Пример латинского плана

Значение фактора В	Значение фактора С			
	С1	С2	С3	С4
<i>B1</i>	A1	A2	A3	A4
<i>B2</i>	A2	A3	A4	A1
<i>B3</i>	A3	A4	A1	A2
<i>B4</i>	A4	A1	A2	A3

Матрица плана дробного факторного эксперимента для $k = 2$

Номер эксперимента	Значение факторов	
	x1	x2
1	0	0
2	0	1
3	1	0
4	1	1

**Матрица плана дробного факторного эксперимента
для $k = 3$**

Номер эксперимента	Значение факторов		
	x_1	x_2	x_3
1	0	0	0
2	0	0	1
3	0	1	0
4	0	1	1
5	1	0	0
6	1	0	1
7	1	1	0
8	1	1	1

§3. Тактическое планирование эксперимента

Совокупность методов установления необходимого объема испытаний относят к тактическому планированию экспериментов.

Поскольку точность оценок наблюдаемой переменной характеризуется ее дисперсией, то основу тактического планирования эксперимента составляют так называемые методы понижения дисперсии. В связи с этим для восприятия последующего материала потребуются некоторые знания математической статистики.

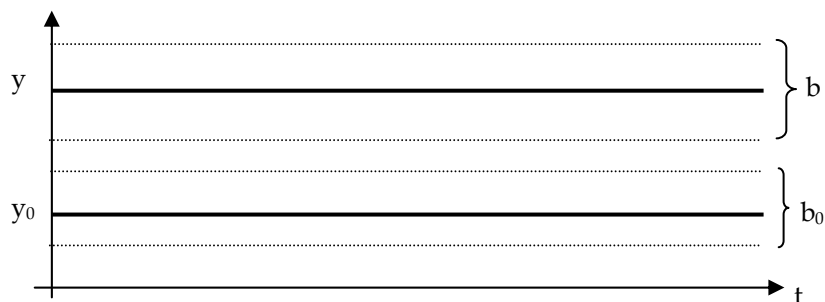
Формирование простой случайной выборки

Поскольку имитационное моделирование представляет собой статистический эксперимент, то при его проведении необходимо не только получить достоверный результат, но и обеспечить его «измерение» с заданной точностью.

Различие понятий «достоверный результат» и «точный результат» можно пояснить с помощью приведенного ниже рисунка. На рисунке использованы следующие обозначения:

y, y_0 – истинное и ошибочное значения наблюдаемой переменной y ;

b, b_0 – доверительные интервалы измерения величин y и y_0 .



В общем случае объем испытаний (величина выборки), необходимый для получения оценок наблюдаемой переменной с заданной точностью, зависит от следующих факторов:

- вида распределения наблюдаемой переменной y (напомним, при статистическом эксперименте она является случайной величиной);
- коррелированности между собой элементов выборки;
- наличия и длительности переходного режима функционирования моделируемой системы.

Если исследователь не обладает перечисленной информацией, то у него имеется единственный способ повышения точности оценок истинного значения наблюдаемой переменной – многократное повторение прогонов модели для каждого сочетания уровней факторов, выбранного на этапе стратегического планирования эксперимента. Такой подход получил название «формирование простой случайной выборки» (ПСВ). Другими словами, при использовании ПСВ каждый «пункт» стратегического плана просто выполняется повторно определенное число раз. При таком подходе общее число прогонов модели, необходимое для достижения цели моделирования, равно произведению $N_C \times N_T$ (N_C – число сочетаний уровней факторов по стратегическому плану; N_T – число прогонов модели для каждого сочетания, вычисленное при тактическом планировании).

Например, если для полного факторного эксперимента $N_C = 64$, а для обеспечения требуемой точности оценок N_T должно быть равно 20, то общее число прогонов модели – 1280. Требуемое время для проведения всех испытаний (по минуте на каждое) – более 20 часов. То есть «модельер» должен трудиться почти сутки без сна и отдыха. Поэтому даже при использовании ПСВ до начала испытаний необходимо определить тот минимальный объем выборки, который обеспечит требуемую точность результатов.

Если случайные значения наблюдаемой переменной не коррелированы и их распределение не изменяется от прогона к прогону, то выборочное среднее можно считать нормально распределенным. В этом случае число прогонов N_T , необходимое для того чтобы истинное среднее наблюдаемой переменной лежало в интервале $y \pm b$ с вероятностью $(1 - \alpha)$, определяется следующим образом:

$$N_T = \frac{Z^2 \cdot D_y}{b^2},$$

где Z – значение нормированного нормального распределения, которое определяется по справочной таблице при заданном уровне значимости $\alpha / 2$;

D_y – дисперсия;

b – доверительный интервал.

Если требуемое значение дисперсии D_y до начала эксперимента неизвестно, целесообразно выполнить пробную серию из L прогонов и вычислить на ее основе выборочную дисперсию, значение которой подставляют в вышеприведенную формулу и получают предварительную оценку числа прогонов модели N_T , затем выполняют оставшиеся $N_T - L$ прогонов, периодически уточняя оценку и число прогонов N_T .

§4. Возможности Matlab/Simulink по планированию и реализации модельных экспериментов

4.1. Разработка планов экспериментов

Планирование экспериментов осуществляется с помощью М-функции Design of Experiments приложения Statistics toolbox.

В состав раздела Design of Experiments входят функции, обеспечивающие разработку всех трех основных видов стратегического плана эксперимента:

- Полного факторного эксперимента (ПФЭ);
- Частичного факторного эксперимента (ЧФЭ);
- Дробного факторного эксперимента (ДФЭ).

Для разработки ПФЭ служит функция *fullfact*. В качестве ее параметров необходимо указать число уровней каждого фактора, участвующего в эксперименте.

Например, если факторы А и В имеют 3 уровня, а фактор С 2 уровня, то обращение к функции *fullfact* выглядит так:

fullfact([3,3,2]).

Введя указанную конструкцию в командном окне *Matlab*, можно получить список всех возможных комбинаций уровней факторов. Список выводится в командном окне, а также сохраняется в рабочей области под именем *ans* (рис. 8.1). Он может быть использован в качестве подсказки либо в текущем сеансе работы с *Matlab*, либо записан в отдельный MAT-файл для последующего применения.

Для формирования плана ДФЭ используется функция *ff2n*. Параметром функции является число факторов. Например, команда *ff2n*(3) обеспечивает вывод в командное окно следующего списка (рис. 8.2).

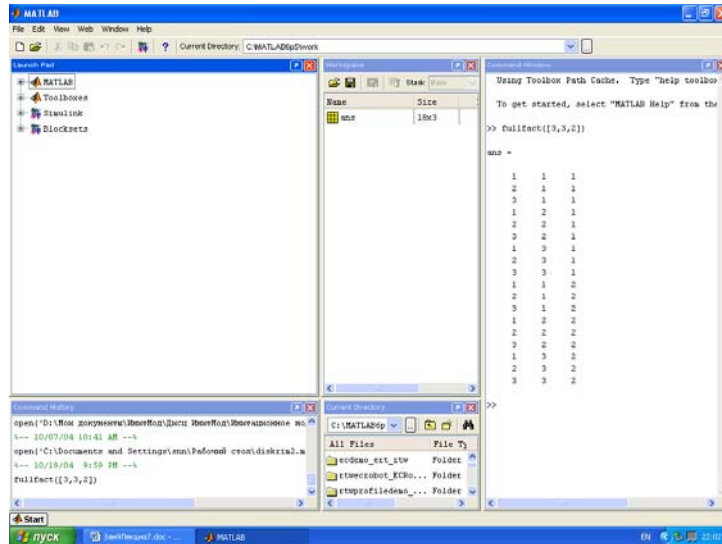


Рис. 8.1. Формирование плана ПФЭ

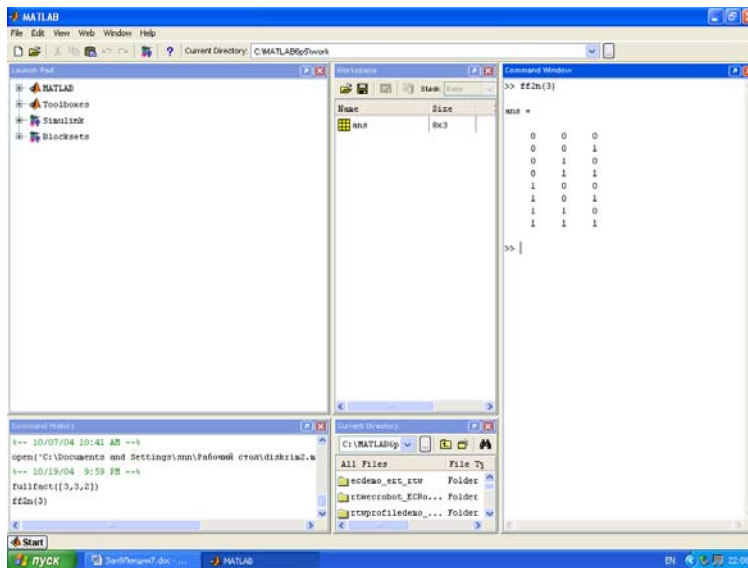


Рис. 8.2. Формирование плана ДФЭ

Формирование ЧФЭ (рандомизированного плана) осуществляется с помощью функции *unidrnd*, представляющей собой генератор дискретной СВ, равномерно распределенной на интервале [1; N]. В общем случае она используется с тремя параметрами

$$\text{unidrnd}(N, k, m),$$

где N – верхняя граница интервала распределения;

k, m – задают число строк и столбцов генерируемой случайной матрицы.

При генерации плана эксперимента эти величины интерпретируются следующим образом:

N – число уровней факторов, участвующих в эксперименте;

k – выбранное пользователем число экспериментов (различных сочетаний уровней факторов);

m – число факторов (рис. 8.3).

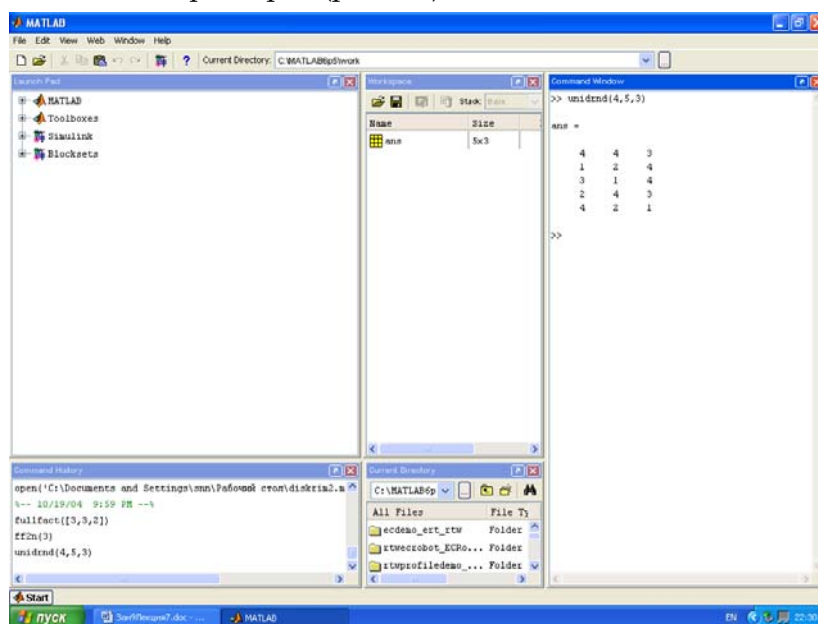


Рис. 8.3. Формирование рандомизированного плана эксперимента

4.2. Проведение имитационных экспериментов с использованием файлов сценариев

Процесс прогона модели при различных значениях параметров (факторов), обработку полученных результатов моделирования можно автоматизировать путем использования файлов сценариев. Matlab располагает механизмом, позволяющим создавать и сохранять устойчивые сценарии в виде специальных М-файлов, которые так и называются – файлы сценариев (Script files), или просто М-сценарии.

М-сценарий представляет собой последовательность команд (или операторов) Matlab, разделенных точкой с запятой (если они записаны в одной строке).

При написании М-сценариев следует учитывать следующее:

- М-сценарий не имеет входных параметров (аргументов);
- М-сценарий может содержать любые М-функции и операторы Matlab;
- Входящие в сценарий М-функции и операторы могут оперировать с данными, находящимися в рабочей области Matlab.

Основным инструментом разработки как М-сценариев, так и М-функций является Редактор/Отладчик Matlab – Editor/Debugger, хотя для этих целей может быть использован любой текстовый редактор.

Для улучшения визуального восприятия текста М-файла его различные компоненты имеют в окне Редактора/Отладчика разный цвет:

- комментарий – зеленый;
- ключевые слова Matlab – синий;
- остальные конструкции – черный.

Порядок использования команд рассмотрим на примере создания сценария, обеспечивающего запуск модели и построение графиков.

1. % Optimal profit tax rate simulation
2. % File: C:\Csr_MtLb\TxRt\TaxRate_DscM.m and TaxRate_Dsc.mdl
3. open_system('TaxRate_Dsc') % Load TaxRate_Dsc.mdl
4. TaxRate=[0:0.05:0.7]% План-вектор эксперимента по ставке налога
5. for Rntb=0.2:0.2:1 % Цикл и план-вектор по рентабельности
6. sim('TaxRate_Dsc')% Run model
7. plot(TaxRate, ScopeData(end,2:end)) %Чертить график поступления в бюджет
8. hold on% Разрешить дополнение графика кривыми
9. grid% Чертить сетку
10. end
11. hold off% Запретить дополнение графика

В m-файле программы за знаком процента всегда идут поясняющие комментарии. Они не являются командами и компьютером не исполняются.

В первой строке программы дается ее назначение, или смысловое название.

Во второй строке – полное имя m-файла, содержащего нашу программу для управления экспериментом над Simulink моделью, и имя файла Simulink модели с расширением .mdl.

Третья строка командой open_system загружает с диска модель в оперативную память.

Четвертая строка присваивает переменной модели TaxRate вектор плана эксперимента по налоговой ставке.

В строках с 5 по 10 выполняется for цикл для проведения экспериментов при различных величинах рентабельности бизнеса.

В шестой строке командой sim запускается модель и начинается моделирование, имитация налогового взаимодействия государства и предприятия.

После окончания имитации команда plot чертит один график, используя данные рабочего (work space) пространства Matlab, записанные туда графопостроителем Scope. Оператор hold on разрешает дополнять рисунок графиками кривых, рассчитанных для других значений циклов рентабельности.

Глава 9. Примеры построения имитационных моделей

§1. Имитационная модель циклов роста и падений в экономике (кризисов)

1.1. Постановка задачи на моделирование

Мировую экономику, экономику отдельных стран и даже отдельных отраслей периодически потрясают кризисы роста и падений. Причины этих явлений до конца не установлены.

Были предположения, что причины кризисов в конкуренции и хаосе. Но это не находило подтверждения. Периодически возникали перепроизводство или же, наоборот, нехватка тех или иных товаров. Исторический опыт показывает, что последние кризисы, как правило, начинались в автомобильной промышленности. Основная причина – сокращение спроса на такой дорогостоящий товар, как автомобили. В то же время исследование причин сокращения спроса – задача не формализуемая и, как следствие, очень сложно решаемая. В этом проблема. Получать количественный прогноз и анализ в таких ситуациях позволяет имитационное моделирование.

Задачей имитационного моделирования с помощью рассматриваемой модели является исследование причинно-следственного механизма возникновения циклов и кризисов перепроизводства.

1.2. Построение концептуальной модели

Предположим, промышленность выпускает автомобили. Существует постоянно растущая потребность в данном товаре. Полагаем, что производство полностью удовлетворяет потребности, но с задержкой (время на разработку новой модели автомобиля, подготовка производства к выпуску и т.д.). Выпущенные автомобили поступают в эксплуатацию от производителей. По-

сколько автомобилей – вещь долговременного пользования, то происходит их накопление. По истечении сроков эксплуатации, износа, аварий и т.д. происходит их выбытие из эксплуатации. Модель должна вычислять предложение на автомобильном рынке, спрос на товар при изменении задержки при производстве автомобилей и тем самым позволять оценить степень устойчивости производства к кризисам и циклам.

1.3. Математическая модель

Имея Simulink с типовыми библиотечными блоками, можно не создавать математическую модель – каждая элементарная модель уже имеет программу, привязанную к блоку. Но составив имитационную модель, можно по этой же схеме составить уравнения для аналитических решений.

Блок-схема имитационной модели кризисов, выполненная в Simulink, представлена на рис. 9.1.

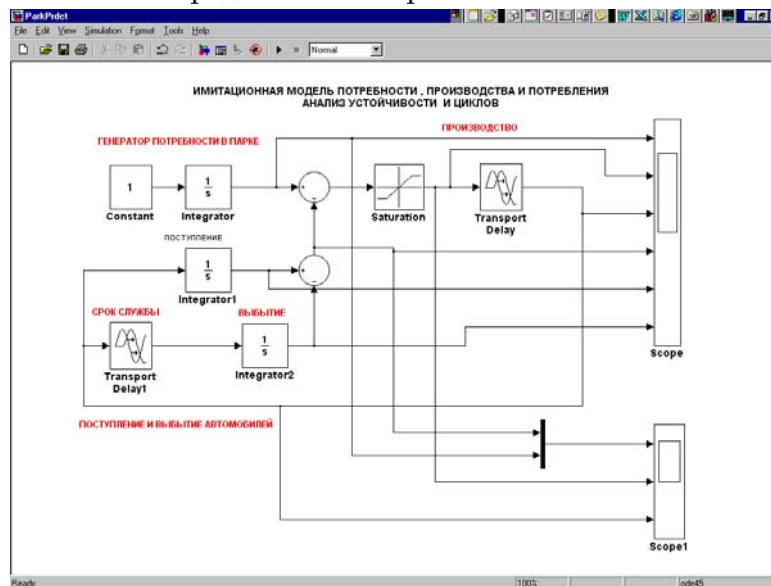


Рис. 9.1. Блок-схема имитационной модели кризисов

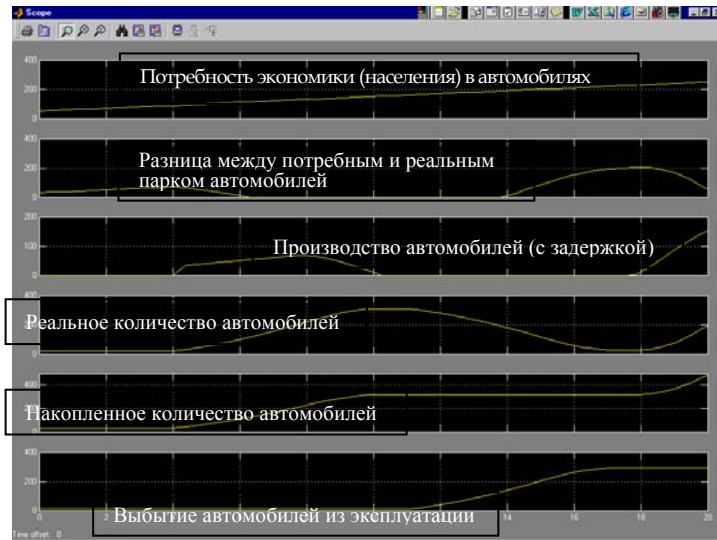
«Генератор потребности в парке» на схеме задается блоком констант и интегратором. На выходе блоков формируется линейная во времени переменная потребности населения в парке автомобилей (необходимый парк автомобилей).

Ниже потребностей расположены четыре блока, отражающие движение парка автомобилей: поступление в эксплуатацию от производителей (вход блока «Срок службы»), накопление их в парке интегратором (блок «Поступление»). Блоки «Срок службы» и «Выбытие» задают поток выбытия автомобилей по ветхости, износу или моральному старению. Нижний круглый блок вычитает из поступившего в парк количества автомобилей выбывшее количество, формируя тем самым реальное количество товаров, находящихся в эксплуатации. Верхний круглый блок сумматора вычитает из необходимого парка наличный парк, создавая переменную текущего спроса. Блоком Saturation (ограничитель) она ограничивается снизу, реализуя традиционную для экономических задач неотрицательность переменных.

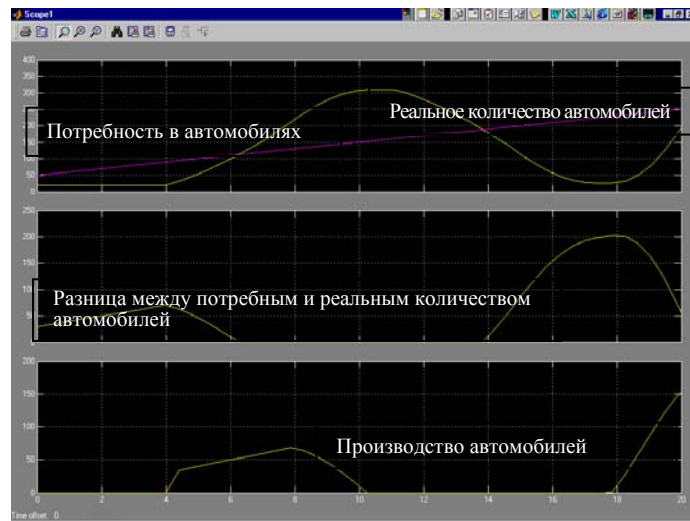
Блок «Производство» задан простейшей моделью, то есть производство выполняет заказ полностью, но с запаздыванием, задаваемым блоком задержки (лаг исполнения заказа).

Параметры в этой модели задаются вручную. Для этого вызывается диалоговое окно настройки параметров – двойным щелчком мыши на блоке. Параметрами, которыми управляется модель, являются:

- задержка в блоке «Производство»;
- задержка в блоке «Службы»;
- начальные условия в интеграторе блока «Поступление» (имитация дефицита автомобилей).



а)



в)

Рис. 9.2. Результаты моделирования (наблюдаемые блоком – а) Score; – в) Score 1)

при следующих исходных данных:

- блок «Constant» Constant value = 10. Значение определяет крутизну прямой потребности населения (экономики) в автомобилях (например, 10 тысяч автомобилей в год);
- блок «Integrator» Initial condition = 50. Значение определяет потребность в автомобилях в момент времени $t = 0$ (например, начальная потребность в автомобилях 50 тысяч);
- блок «Saturation» Upper limit = 1000, Lower limit = 0. Первое значение определяет ограничение сверху, второе ограничение снизу. Это необходимо для исключения отрицательных значений, что характерно для экономических задач;
- блок «Transport delay» Timer delay = 4. Значение определяет задержку во времени производства автомобилей относительно возникшей потребности в них (например, 4 года);
- блок «Integrator 1» Initial condition = 30. Значение определяет начальный уровень накопленного парка автомобилей (например, 30 тысяч штук);
- блок «Integrator 2» Initial condition = 10. Значение определяет начальный уровень количества автомобилей выбывающих из строя по причине старения, аварий и т.д. (например, 10 тысяч штук);
- блок «Transport delay 1» Timer delay = 7. Значение определяет срок эксплуатации автомобилей (например, 7 лет).

§2. Использование имитационного моделирования для поиска оптимальной ставки налогообложения на прибыль

2.1. Постановка задачи на моделирование

Государство стремится увеличить налоги, чтобы наполнить бюджет для выполнения своих социально-экономических и оборонных функций. Производители товаров и услуг (бизнес) жалуются, что налоговое бремя велико, и считают, что налоговые ставки надо уменьшить.

Специалисты (экономисты) утверждают, что большие налоги сдерживают развитие экономики, а значит, и будущие наполнение бюджета. Проблема состоит в том, чтобы определить оптимальную ставку налогообложения. При этом исходят из того, что поступления в бюджет за определенный период времени будут наибольшими не при максимальной, а при оптимальной для бюджета ставке налога. То есть с налоговой ставки поступления в бюджет будут увеличиваться до определенного предела, а затем уменьшаться.

Цель моделирования состоит в том, чтобы исследовать зависимость поступлений в бюджет от величины налоговой ставки и обосновать величину налоговой ставки.

2.2. Построение концептуальной модели

Несмотря на массу существующих налогов, источником развития производства (бизнеса) и источником налогового пополнения бюджета в конечном счете является прибыль, т.е. превышение доходов над расходами.

Ставка налога объявляется законодательно. Бюджет получает налоговые отчисления от прибыли предприятий.

Таким образом, описательная модель выглядит следующим образом.

Государство объявляет ставку налога на прибыль и получает от предприятий (фирм) средства в бюджет. Предприятия (фирмы) обладают собственным капиталом, производят прибыль, отчисляют по налоговой ставке средства в бюджет. Постналоговая прибыль как нераспределенная прибыль полностью включается в собственный капитал предприятия (фирмы). Принимаем, что при моделировании дивиденды не выплачиваются, никаких других отчислений от прибыли не производится. Вся прибыль распределяется только на два потока:

- в бюджет;
- в собственный капитал предприятия (фирмы).

Графически концептуальная модель выглядит следующим образом (рис. 9.3).

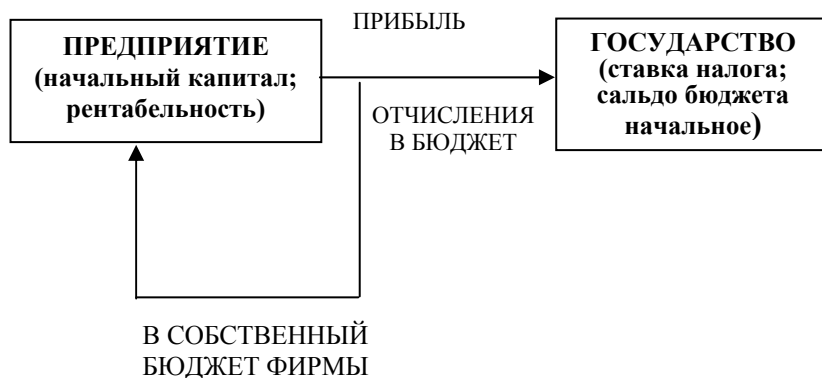


Рис. 9.3. Концептуальная модель

2.3. Математическая модель

Сумма налоговых поступлений от предприятий в бюджет за моделируемый период определяется интегралом

$$BD(t) = \int_{t=tb}^{t=tf} PRF(t) \cdot TXRT \cdot dt,$$

где $BD(t)$ – сумма поступивших в бюджет средств от начала моделирования к моменту t , руб.;

$PRF(t)$ – доналоговая прибыль (profit), получаемая предприятием в момент t , руб./год;

$TXRT$ – ставка налога на прибыль (tax rate);

T – текущее время;

tb – начальный момент моделирования (begin);

tf – последний момент моделирования (final).

Капитализируемый предприятием за время моделирования остаток прибыли:

$$CP(t) = \int_{t=tb}^{t=tf} PRF(t) \cdot (1 - TXRT) \cdot dt.$$

Прибыль в момент t :

$$PRF(t) = CP(t) \times RN,$$

где RN – рентабельность капитала предприятия.

Задается как параметр предприятия, исходное данное.

Итак, исходными данными для моделирования являются:

- налоговая ставка (ее надо оптимизировать);
- рентабельность;
- начальный капитал предприятия;
- интервал моделирования.

При моделировании предполагается устанавливать для предприятий с различным уровнем рентабельности различные ставки налога и измерять поступления в бюджет. На основе полученных данных будет выбираться та ставка налога, которая обеспечивает максимальные поступления в бюджет.

2.4. Компьютерная модель в программе Simulink

Имитационную модель системы налогообложения можно представить в виде блок-схемы, содержащей типовые функциональные блоки (рис. 9.4).

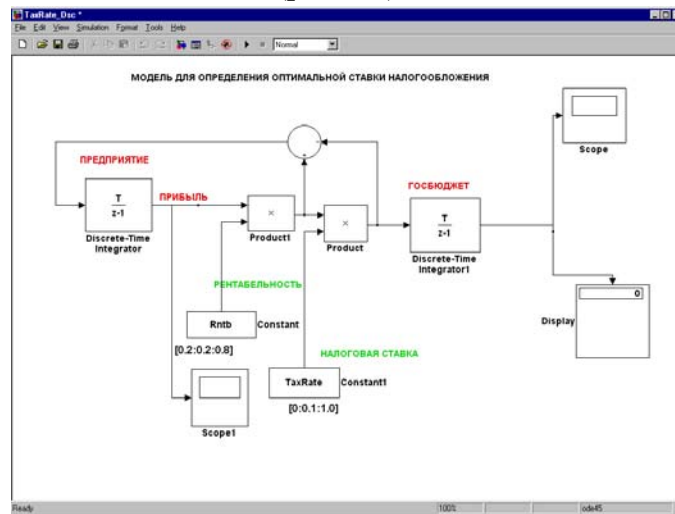


Рис. 9.4. Блок-схема имитационной модели для определения оптимальной ставки налогообложения

Блок «Предприятие» представляет накопитель собственного капитала предприятия, из библиотеки элементов он взят как блок «Интегратор дискретного времени». На вход блока поступает постналоговая, нераспределенная прибыль. Эта прибыль накапливается, увеличивая собственный капитал предприятия. Выход блока – величина капитала предприятия. В блоке задается начальный капитал.

Блок умножения производит умножение полученного капитала на рентабельность. Рентабельность задается библиотечным блоком «Константа» с именем Rntb. Комментарий под блоком задает вектор плана эксперимента по фактору рентабельности [0.2:0.2:0.8]. Последнее означает, что имитационные эксперименты будут проводиться для рентабельности от 20 до 80% с шагом 20%. Для автоматизации экспериментов константу рентабельности мы можем заменить на переменную и управлять ее значениями из программного файла Matlab.

Блок «Score 1» служит для отображения переменной капитала предприятия.

Следующий блок умножения формирует произведение прибыли предприятия на налоговую ставку TaxRate, т.е. поток отчислений от прибыли в госбюджет.

Ставка налога задается библиотечным блоком «Константа». Под блоком как комментарий задан вектор плана экспериментов по фактору налоговая ставка [0:0.1:1.0]. Последнее означает, что имитационные эксперименты будут проводиться для ставки от 0 до 100% с шагом 10%. Для автоматизации имитационных экспериментов константу налоговой ставки заменяем на переменную.

Блок «Госбюджет» представлен интегратором. Он аккумулирует налоговые поступления за период моделирования.

Блоки «Score» и «Display» отображают график и числовые значения накопления средств от налога в бюджете соответственно.

Круглый блок вычисляет чистую прибыль предприятия как разницу между доналоговой прибылью и частью прибыли, отчисляемой по налоговой ставке в бюджет.

2.5. Исходные данные для параметров, переменных и показателей модели

В качестве исходных данных задаются числовые значения:

- налоговой ставки;
- рентабельности;
- начального капитала предприятия;
- интервал моделирования.

2.6. Математическая схема модели и метод решения

Для решения данной задачи используются непрерывно-детерминированная модель (D-схема). Выполняется имитационное моделирование процесса развития предприятия и накопления налоговых средств в бюджете во времени решением системы дифференциальных уравнений стандартными средствами Matlab и Simulink.

2.7. Средства управления экспериментом

Средствами отображения результатов моделирования являются графопостроители Score и индикаторы чисел Display.

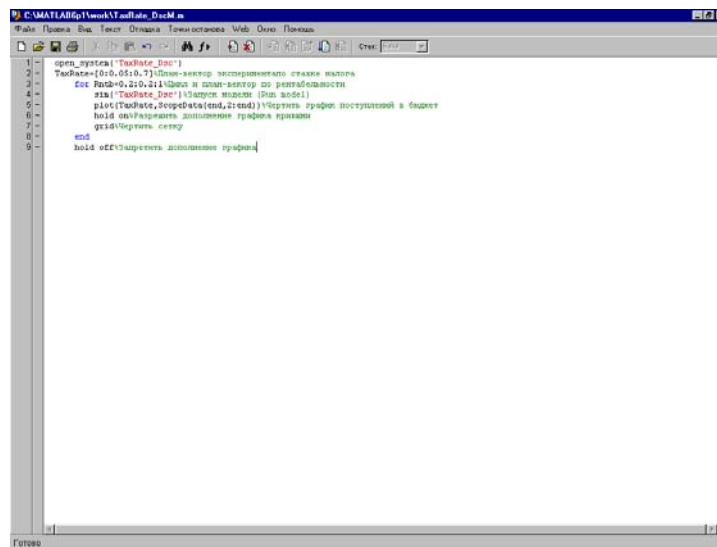
Средствами управления экспериментом являются диалоговые окна констант-факторов:

- ставка налога;
- рентабельность.

Двойным щелчком мыши открываются окна и меняются значение фактора, устанавливая тем самым для предприятия ставку налога. Запуская модель, оценивают поступления в бюджет. Это так называемое ручное управление модельным экспериментом. Для автоматизации планирования и управления экспериментом составляется программа на языке Matlab в файле с расширением .m.

2.8. Программа управления имитационным экспериментом

Ниже представлен вариант программы планирования и управления экспериментом (рис. 9.5).



```
1 open_system('TaxRate_000')
2 TaxRate=[0.0,0.0,0.7] %Вектор экспериментальной ставки налога
3 for Rate=0.1:0.2:1 %Шаг и цикл-вектор по рентабельности
4     sim('TaxRate_000') %Запуск модели (для Rate=0)
5     plot(TaxRate,ScoreData(end,:)) %Чертит график поступления в бюджет
6     hold on %Удерживает диспозицию графика кривой
7     axis([0 100 0 100])
8 end
9 hold off %Удерживает диспозицию графика
```

Рис. 9.5. Программа управления экспериментом

В m-файле программы за знаком процента всегда идут поясняющие комментарии. Они не являются командами и компьютером не исполняются.

Первая строка командой `open_system` загружает с диска модель в оперативную память.

Вторая строка присваивает переменной модели `TaxRate` вектор плана эксперимента по налоговой ставке.

В строках с 3 по 9 выполняется `for` цикл для проведения экспериментов при различных величинах рентабельности предприятия.

В четвертой строке командой `sim` запускается модель и начинается моделирование, имитация налогового взаимодействия государства и предприятия.

После окончания имитации команда `plot` чертит один график, используя данные рабочего (`work space`) пространства Matlab, записанные туда графопостроителем `Score`. Оператор `hold on` разрешает дополнять рисунок графиками кривых, рассчитанными для следующих значений циклов рентабельности.

§3. «Паутинообразная» модель фирмы (равновесие на конкурентном рынке)

3.1. Постановка задачи на моделирование

Основоположник ценовой теории Альфред Маршалл (1842–1924) полагал, что большинство экономических процессов можно объяснить терминами равновесной рыночной цены. Цена устанавливается при взаимодействии спроса и предложения.

Обычно на бумаге или доске чертят пересечение линий спроса и предложения в зависимости от цены товара. Смещают линии, меняют их крутизну, наблюдают точки новых равновесий. Объясняют ножницы дефицита, инфляцию, перепроизводство и др. Все это можно делать с использованием Matlab.

На уровне фирмы задача может выглядеть следующим образом.

Предприниматель собирается вложить средства в создание фирмы, которая будет выпускать товар и реализовывать его на рынке. Его интересует, как будет вести себя цена на товар при изменении объема производства. Опыт подсказывает, что при увеличении производства происходит падение спроса и приходится снижать цену. Необходимо получить ответ, при каких условиях цена будет стабильной. Подобная задача может быть решена с использованием имитационного моделирования.

В литературе описано несколько вариантов такой модели. Все они обладают определенными одинаковыми свойствами. Обычно в них предполагается, что спрос на некоторый продукт (чаще всего рассматривается сельскохозяйственная продукция) на заданном отрезке времени зависит от цены

(и других факторов) на этом отрезке. Что же касается предложения, то оно определяется ценами предыдущего периода времени (недели, месяца, квартала и т.д.). Кроме того, предполагается, что рынок всегда находится в условиях локального равновесия. Исторически такая модель получила название «паутинообразной». Почему она так названа, будет ясно из результатов моделирования.

Существуют четыре варианта этой модели: детерминированная, вероятностная, модель с обучением и модель с запасами.

В детерминированной модели отсутствует учет влияния случайных факторов. В вероятностной модели учитываются влияние на спрос непредвиденных колебаний предпочтений и доходов потребителей, а также другие случайные факторы, влияющие на величину спроса. Предложение на предыдущем отрезке времени также считается подверженным влиянию случайных факторов. Они отражают влияние колебаний технологии и эффективности производственного процесса и т.д. Наконец, условие локального равновесия означает совпадение спроса и предложения с точностью до некоторой случайной величины.

В модели с обучением предполагается, что поставщики учитывают сложившуюся тенденцию изменения цен и с учетом этого планируют выпуск продукции на очередной отрезок времени.

В последних двух моделях цены устанавливаются на таком уровне, чтобы обеспечить локальное равновесие рынка только за счет текущего производства, и никаких запасов продукции не создается (например, потому, что продукты быстро портятся).

В модель с запасами вводится дополнительная группа участников рыночного механизма, которых можно назвать «коммерсантами». Они держат запасы и организуют торговлю.

Для нашего случая больше подойдет вероятностная модель с обучением. При каких допущениях она составлена? Как выглядит зависимость для определения текущего спроса?

Предполагается, что спрос на T -м отрезке времени линейно зависит от текущей цены и, кроме того, спрос подвер-

жен случайному разбросу. Таким образом, для описания спроса используется зависимость спроса от цены

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

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

D_0 – спрос при нулевой цене;

K_d – крутизна линии спроса;

P_{rc} – подлежащая определению цена (price) на T -м отрезке времени;

U – случайная величина с заданным законом распределения.

Полагаем, что спрос симметрично колеблется относительно среднего значения, которое определяется постоянными коэффициентами линейного уравнения. Поэтому можно выбрать нормальное распределение с нулевым математическим ожиданием и заданным средним квадратическим отклонением (СКО).

Предложение на текущем отрезке также линейно зависит от цены, но не текущей, а представляющей собой некоторую комбинацию цен на двух предыдущих отрезках времени. В простейшем случае это может быть средняя цена. Поэтому для расчета предложения используется следующая зависимость:

$$S_{pl} = S_0 + K_s * P_{rc} + V ,$$

где S_{pl} – предложение (supply) на T -м отрезке времени;

S_0 – предложение при нулевой цене;

K_s – крутизна линии предложения;

P_{rc} – подлежащая определению цена (price) на T -м отрезке времени;

V – случайная величина с заданным законом распределения.

Наконец, условие локального равновесия рынка, которое можно записать так:

$$S_{pl} = D_{md} + W , \tag{1}$$

где W – случайная величина с заданным распределением.

Случайная величина W характеризуется нулевым математическим ожиданием и средним квадратическим отклонением.

Учитывая то, что модель упрощенная, этап постановки задачи и этап построения концептуальной модели совпали.

Подставляя выражения для D_{md} и S_{pl} в (1) и разрешая уравнение относительно P_{rc} , получаем:

$$S_{pl} = D_0 - K_d * P_{rc} + U + W,$$

$$P_{rc} = (D_0 - S_{pl}) / K_d - U - W.$$

Задача моделирования заключается в исследовании влияния параметров системы на характер зависимости цены от времени.

3.2. Построение модели

Блок-схема имитационной модели в Simulink выглядит следующим образом (рис. 9.6).

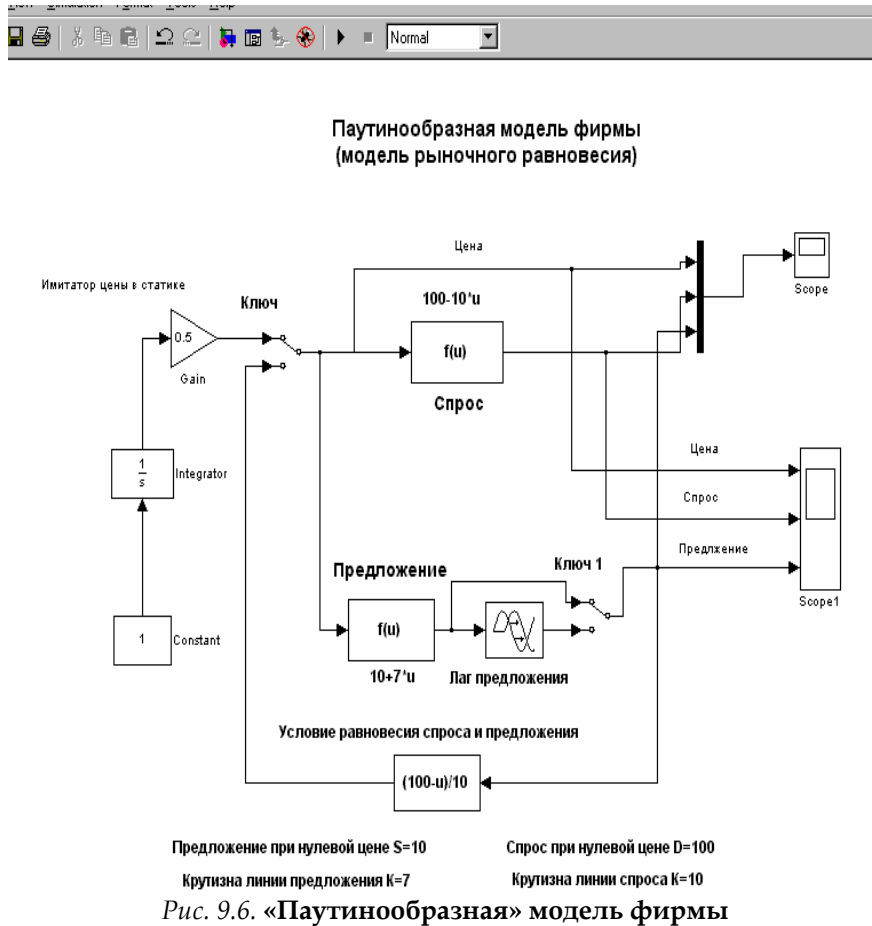
Экономическое содержание модели представляют лишь четыре блока, расположенные в центре окна.

Спрос представлен одним стандартным блоком. Он вычисляет значение спроса в зависимости от цены, подаваемой на вход блока. Там же учитывается влияние случайных факторов.

Обозначения и параметры блока на схеме следующие:

$$U = P_{rc}, \quad D_0 = 100, \quad K_d = 10.$$

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



Обозначение и параметры блока на схеме следующие:

$$U = P_{тс}, \quad S_0 = 10, \quad K_s = 7.$$

Блок «Лag» имитирует запаздывание поставщика на рынке. Продавец поставляет товар в количестве, определенном на основе цен прошлого интервала времени.

Блок «Равновесие» имитирует решение поставщика смириться с ценой текущего спроса. Он соглашается продать весь товар по цене, которую диктует линия спроса.

Практикум

Практическое занятие 1

Основы работы в MATLAB/SIMULINK

Задание 1

1. Произведите запуск MATLAB. Ознакомьтесь с интерфейсом программы.
2. Произведите запуск Simulink. Ознакомьтесь с окном браузера библиотеки (Simulink Library Browser). Выполните в браузере команду File – New – Model, создайте пустое окно блок-диаграммы модели (untitled).
3. Создайте первую модель в соответствии с приведенными инструкциями.

Постановка задачи.

Предположим, что интересующая вас информация может находиться в интернете на одном из двух сайтов (Сайт 1 и Сайт 2). Обнаружив искомую информацию, вы скачиваете ее на свой компьютер; если информация имеется на обоих сайтах, то в качестве источника выступает Сайт 1 (будем считать, что он отличается лучшей организацией данных). Предположим также, что в любом случае вы должны сообщить о результатах поиска своему шефу. Такая ситуация описывается с помощью детерминированного автомата. Обобщенную модель конечного детерминированного автомата в Matlab описывает блок Combinatorial Logic (раздел Logic and Bit Operations библиотеки Simulink). Блок имеет единственный параметр настройки – Truth table (таблица истинности), который представляет собой список возможных значений автомата (рис. 1.1).

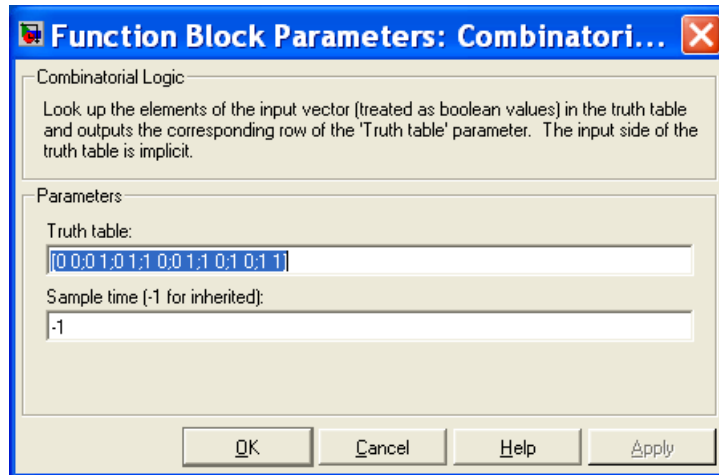


Рис. 1.1. Окно настройки блока Combinatorial Logic

Таблица истинности для приводимого примера имеет вид (табл. 1.1).

Таблица 1.1

**Описание поиска информации в интернете
в виде таблицы истинности**

Входной сигнал (наличие информации)		Выходной сигнал (результат поиска)		
Сайт 1	Сайт 2	Загрузка данных с Сайта 1	Загрузка данных с Сайта 2	Сообщение шефу
0	0	0	0	1
0	1	0	1	1
1	0	1	0	1
1	1	1	0	1

Для рассматриваемой ситуации значение параметра Truth table будет выглядеть следующим образом:
[001; 011; 101; 101]

Создание модели

Используя блоки Combinatorial Logic, Constant, Display и Mux, соберите схему модели (рис. 1.2). Чтобы создаваемая модель более наглядно отражала существо рассматриваемой задачи, замените метки блоков введенными обозначениями: Посетитель, Сайт 1, Сайт 2.

Измените таблицу истинности блока Посетитель в соответствии с логикой его поведения для различных значений входного сигнала (рис. 1.3).

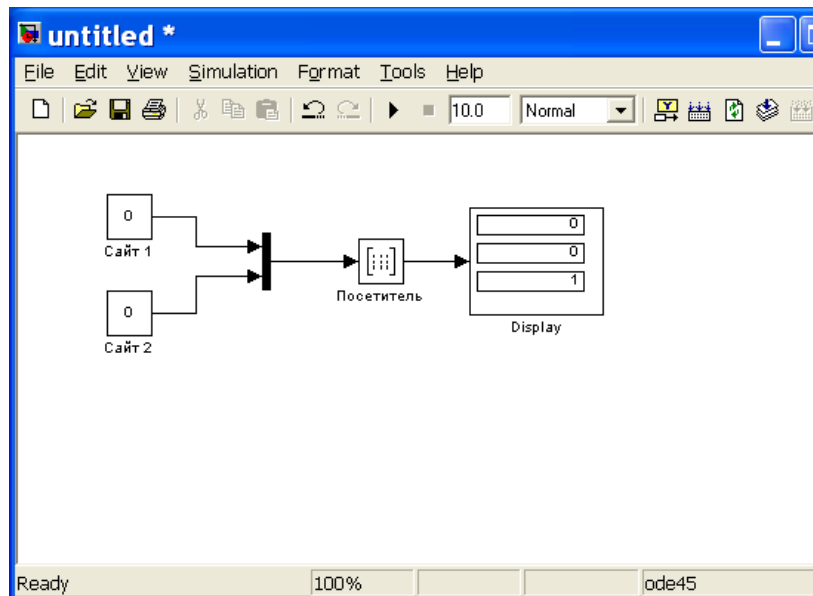


Рис. 1.2. Блок-диаграмма первой модели

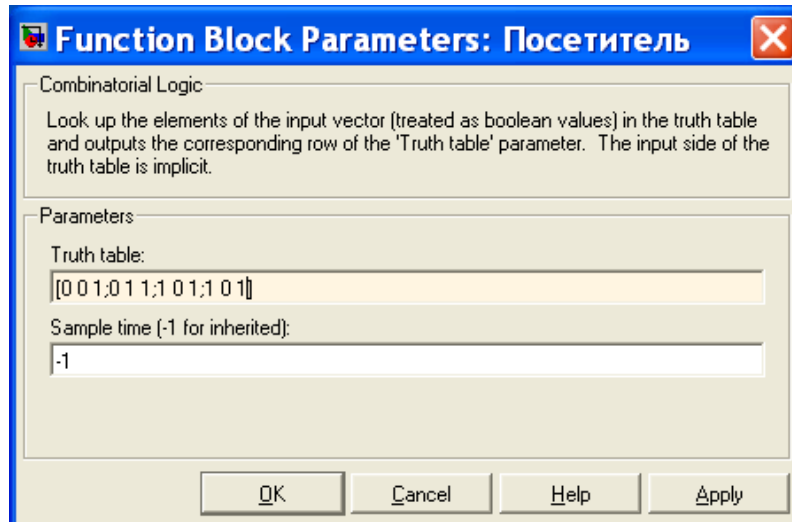


Рис. 1.3. Установка нового значения параметра Truth table

Проведение имитационных экспериментов

Установите значения констант Сайт 1 и Сайт 2 равным нулю (то есть считается, что ни тот, ни другой узел не содержат требуемой информации). Запустите модель на исполнение.

Что вы наблюдаете в блоке Display? Что означает эта информация? Ответы оформите в отчет для отправки преподавателю.

Измените значения констант блоков Сайт 1 и Сайт 2 в соответствии с табл. 1.1 и проведите запуск модели. Что вы наблюдаете? Что означают показания блока Display? Впишите в отчет.

(Ответы на все задания оформляйте в Word и по окончании практического занятия, используя меню Файлы в Прометее отправьте преподавателю. Зачет по каждому практическому занятию выставляется при наличии отчета и сданном тесте).

Задание 2

1. Доработайте модель блоком To Workspace (раздел библиотеки *Sinks*) для сохранения результатов моделирования (рис. 1.4).

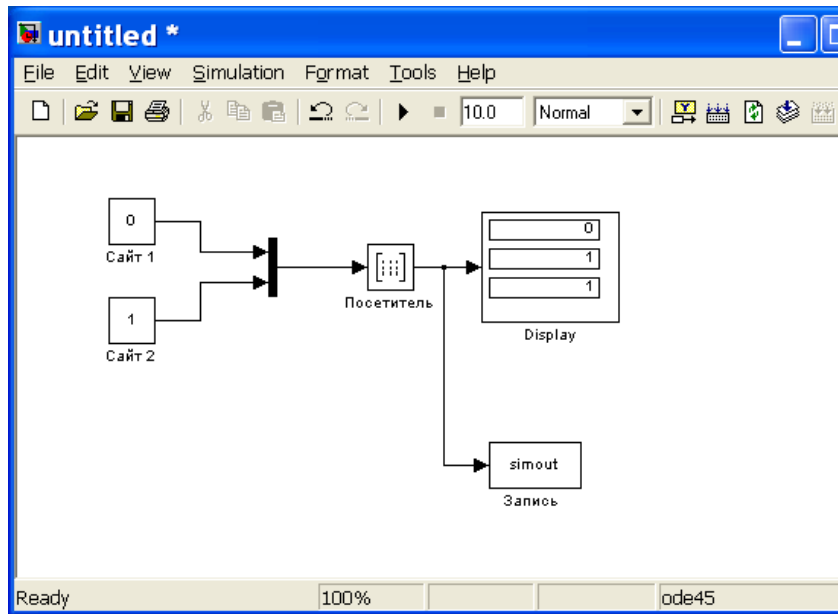


Рис. 1.4. Сохранение результатов моделирования с помощью блока To Workspace

а) Установите параметры моделирования (меню Simulation) - Fixed-step, discrete (no continuous states, Stop time = 1, Fixed step size = 1).

б) Запустите модель на исполнение. После сеанса моделирования откройте командное окно Matlab и в командной строке наберите имя переменной Simout и нажмите Enter. В качестве ответа Matlab выведет в окно содержимое матрицы Simout (рис. 1.5).

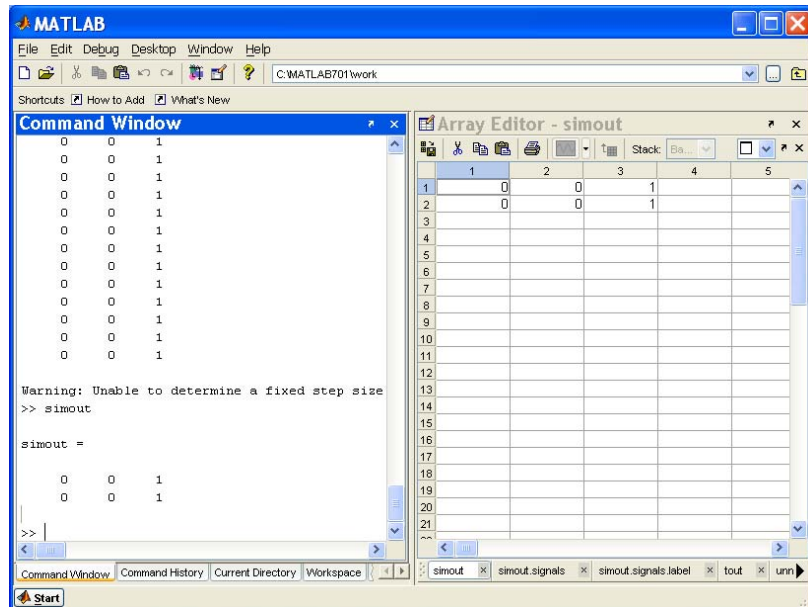


Рис. 1.5. Вывод результатов моделирования в командном окне

в) Для исключения дублирования результатов моделирования (см. рис. 1.5) при сохранении их в рабочей области установите значение Stop time, равное нулю (меню Simulation). Прделайте п. б). Что вы наблюдаете и почему? Впишите в отчет.

г) Проведите имитационные эксперименты с моделью, сохраняя результаты всех экспериментов в рабочей области. Для чего:

- установите первую пару значений констант Сайт 1 и Сайт 2 в соответствии с табл. 1.1. (0,0). Выполните эксперимент, откройте окно рабочей области и убедитесь, что регистрация прошла успешно;
- перед следующим запуском модели внесите в нее следующие изменения:
 - установите новое значение одной из констант в соответствии с табл. 1.1,

- замените имя матрицы регистрации **Simout** на **Simout1** (в окне настроек блока to Wotkspace);
 - изменяя аналогичным образом значения констант и имя матрицы регистрации (Simout2, Simout3), выполните оставшиеся эксперименты.
- д) Запишите в отчет, что вы наблюдаете в окне рабочей области.

Практическое занятие 2

Моделирование случайных событий и величин

1. Моделирование случайных событий

1.1. Моделирования простого события

Задание 1

1. Используя инструментарий Simulink, выполнить модель простого события А, вероятность появления которого равна 0,4. Схема данной модели приведена на рис. 2.1.

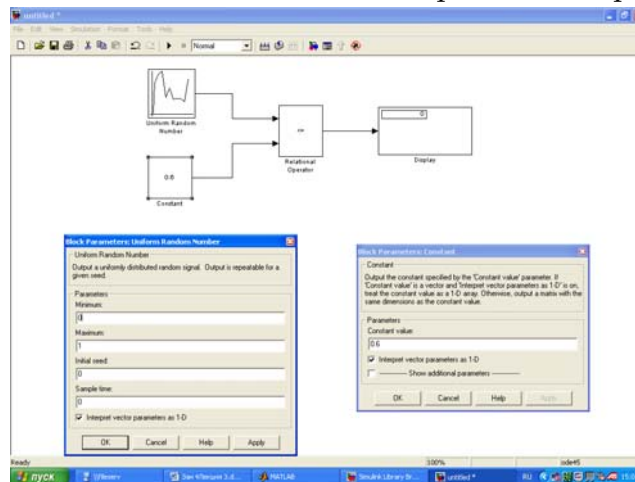


Рис. 2.1. Блок-схема модели

2. Пояснить назначение каждого блока модели и параметров блоков.
3. Произвести однократный пуск модели, путем нажатия кнопки ►. Ответить на вопрос: что регистрирует блок Display?
4. Производя многократный запуск модели, наблюдать за показанием блока Display. Почему наблюдается такие показания? Какой параметр и в каком блоке надо изменить, чтобы менялись показания регистрирующего блока?
5. Укажите, каким недостатком обладает эта модель?

(Ответы на все задания оформляйте в Word и по окончании практического занятия, используя меню Файлы в Прометее, отправьте преподавателю. Зачет по каждому практическому занятию выставляется при наличии отчета и сданном тесте).

Задание 2

1. В соответствии с рис. 2.2 выполнить модель простого события А, вероятность наступления которого равна 0,3.

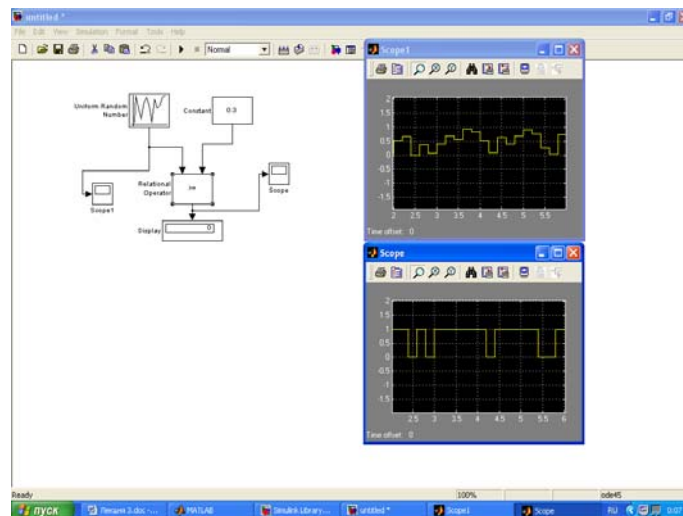


Рис. 2.2. Блок-схема модели

2. Произвести однократный пуск модели. Ответить на вопрос: что регистрируют блоки Score, Score1, Display?

3. Что вы наблюдаете при многократном пуске модели и почему?

4. Измените вероятность появления события на 0,8. Что произойдет с показаниями блока Score? Почему?

5. Доработайте модель путем замены блока Uniform Random Number на блок Fcn (раздел Function&Tables), задав в качестве параметра блока функцию из раздела Toolboxes-Statistics (Средства статистического анализа) – Random Number Generation (генераторы случайных чисел). При этом выберите ГСЧ, формирующий равномерно распределенные случайные числа в интервале (0,1).

Напоминание по использованию раздела Toolboxes-Statistics (Средства статистического анализа)-Random Number Generation (генераторы случайных чисел).

- 1) Открыть встроенную справочную систему Matlab (раздел Toolboxes-Statistics).
- 2) В списке Random Number Generation выбрать функцию, соответствующую требуемому закону распределения.
- 3) Двойным щелчком ЛКМ на выбранной строке открыть страницу справочника, содержащую описание данного генератора; при этом в верхнем левом поле окна будет выведено название генератора; выделите его с помощью мыши и скопируйте в буфер обмена (используя комбинацию клавиш <Ctrl> + <C>).
- 4) В блок-диаграмме выбрать блок, в котором будет использоваться генератор, и открыть окно его настроек.
- 5) Вставить из буфера обмена название генератора (сочетание клавиш <Ctrl> + <V>).

Ввести требуемые значения параметров «запуска» генератора.

6. Проводя моделирование, ответьте на вопрос: чем отличаются показания регистрирующих блоков в данной модели и предыдущей? Почему?

1.2. Моделирование полной группы несовместных событий

Задание 3

1. Самостоятельно разработать схему моделирования ПГНС, для следующих исходных данных: ПГНС состоит из трех независимых событий: A_1, A_2, A_3 ; вероятности появления соответственно: $P = 0,2; P_2 = 0,7; P_3 = 0,1$.

2. Регистрацию появления событий производить с использованием блоков Scope и Display.

3. Описать, что Вы наблюдаете на регистрирующих блоках при пусках модели.

4. Измените параметры модели так, чтобы одна пара регистрирующих блоков Scope и Display показывала при каждом пуске модели нули. Ваше решение впишите в отчет.

2. Моделирование случайных величин

2.1. Моделирование непрерывных случайных величин

Задание 4

1. Используя Demos раздела Toolboxes-Statistics (Средства статистического анализа) – Random Number Generation (генераторы случайных чисел), выбрать ГСЧ, распределенных по нормальному закону (Normal).
2. Как называются параметры Mu и Sigma и что ими задается?
3. Измените значение Sample, установив 100, 1000, 10000, 100000. Что вы наблюдаете и как это объяснить?
4. Приведите примеры случайных величин, распределенных по нормальному закону.
5. Выберите ГСЧ, распределенных по экспоненциальному закону. Что задается параметром Mu?
6. Приведите примеры случайных величин, распределенных по экспоненциальному закону.
7. Выполните модель в соответствии со следующим заданием (рис. 2.3). Требуется сравнить эффективность использова-

ния двух топливозаправочных станций (ТЗС) на протяжении 11 дней. В качестве показателей эффективности используется коэффициент оборудования топливозаправочных станций – $K_{и}$. Эта величина рассчитывается как отношение количества заправленных машин к потенциально возможной пропускной способности станций.

Исходные данные:

- Первая топливозаправочная станция имеет 8 топливозаправочных колонок (ТЗК).
- Вторая топливозаправочная станция имеет 4 ТЗК.
- Среднее время заправки одного автомобиля на любой из заправок составляет 5 мин.
- Работа ТЗС круглосуточная (24 часа).
- Количество автомобилей, заправленных в течение суток, – величина случайная и подчиняется нормальному закону распределения. Для первой ТЗС закон распределения СВ имеет параметры $m_1=1000$ авто, $v_1=250$ авто; для второй ТЗС: $m_2=850$ авто, $v_2=70$ авто.

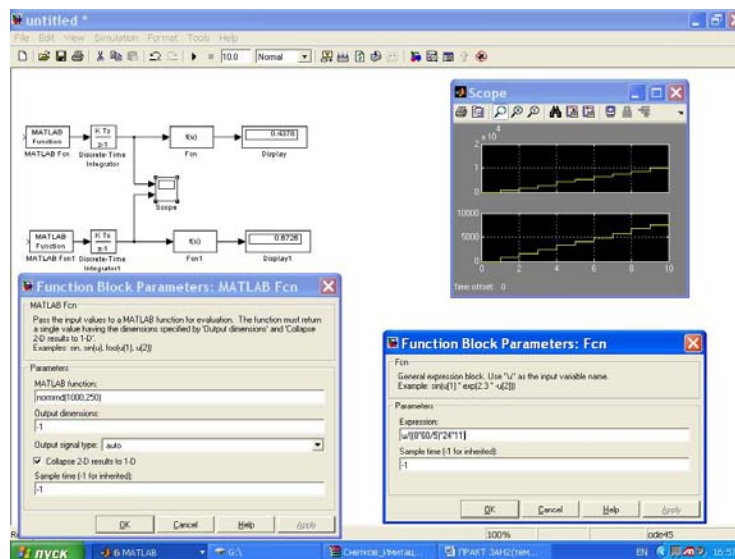


Рис. 2.3. Блок-схема модели

8. Поясните назначение каждого блока модели и заданные параметры.
9. Интерпретируйте результаты моделирования.

Практическое занятие 3

1. Управление модельным временем

Задание 1

1. Используя инструмент Simulink, выполнить модель потока заявок на обслуживание при следующих исходных данных: среднее время между заявками на обслуживание $T_{cp} = 1$ мин.; закон плотности распределения – экспоненциальный.

Продвижение модельного времени задать с фиксированным шагом Fixed step (задается в разделе параметры моделирования). Схема модели приведена на рис. 3.1.

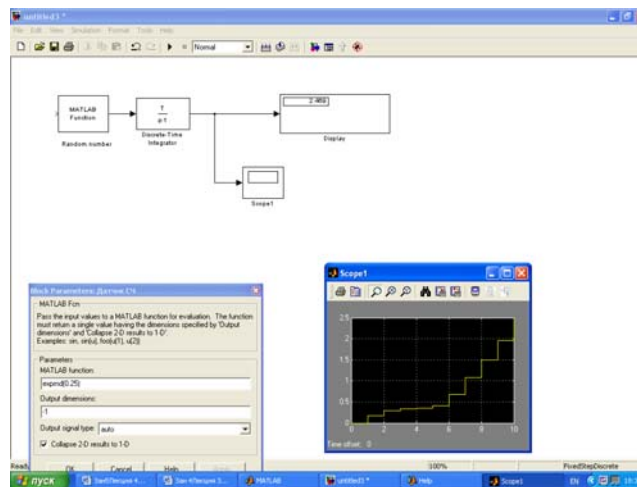


Рис. 3.1. Схема модели потока заявок

2. Пояснить назначение каждого блока модели и параметров блоков.
3. Произвести однократный пуск модели, путем нажатия кнопки ►. Ответить на вопрос: что регистрирует блок Display?
4. Что наблюдается на экране блока Score (по горизонтали и по вертикали)?

Задание 2

1. Выполнить модель потока заявок на обслуживание (например, посетителей супермаркета) с обслуживанием заявок (под обслуживанием в данной задаче понимать прием денег в кассу) (рис. 3.2). Исходные данные для моделирования:
 - Закон плотности распределения интервалов между заявками – экспоненциальный.
 - Среднее время между заявками (интервал между покупателями) $T_{cp} = 1$ мин.
 - Закон плотности распределения суммы покупки – нормальный с параметрами $МОЖ = 200$ руб., $СКО = 50$ руб.
 - Время подсчета покупки принять равным «0».
2. Установить в параметрах моделирования переменный шаг моделирования.
3. Цель моделирования с помощью данной модели – прогнозирование доходов в супермаркете.

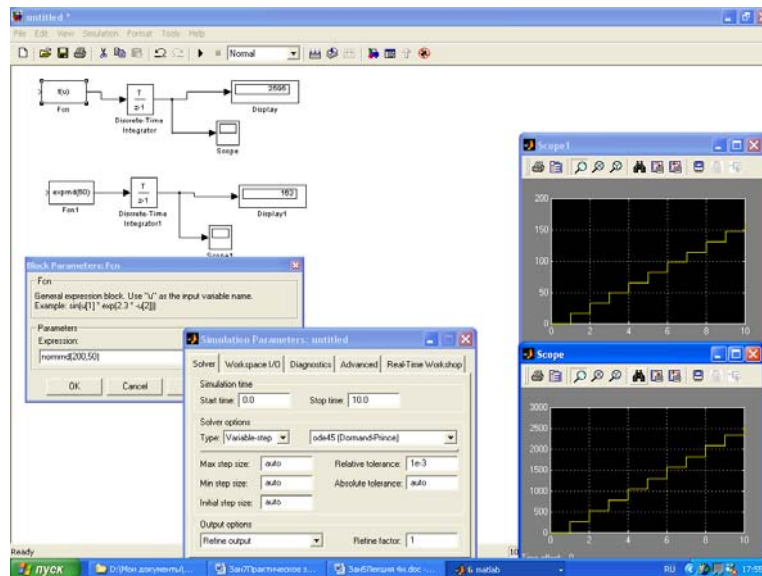


Рис. 3.2. Модель потока заявок с обслуживанием (время обслуживания равно «0»)

4. Что имитирует нижняя часть модели?
5. Что имитирует верхняя часть модели?
6. Что регистрируют блоки Display, Scope?
7. Что задается параметрами блоков параметры блока F_{ch} ?
8. Задать фиксированный шаг моделирования фиксированный шаг, как изменились результаты моделирования и почему?

2. Моделирование синхронных процессов

Задание 3

1. Собрать модель двух синхронных процессов (рис. 3.3):
 1-й процесс – процесс обслуживания заявки (процесс оплаты в кассе, при этом время обслуживания не равно «0», данный процесс подчиненный по отношению ко второму процессу).

2. Провести моделирование, по результатам моделирования описать, что регистрируют контролирующие блоки Scope, Scope1, Scope2, Scope3, Scope4, Display, Display1.
3. Для чего предназначены блоки Hit Crossing и сумматор?

Задание 4

1. Доработать предыдущую имитационную модель для случая двухканальной системы обслуживания (например, две кассы). Окончание моделирования выполнить по условию просмотра 300 единиц модельного времени. Среднее время обслуживания в 1-й кассе (кассир более опытный и расторопный) $T_{cp} = 1$ мин., во 2-й кассе 5 мин. Интервал между покупателями $T_{cp} = 0,5$ мин. для обеих касс. Остальные условия такие же, как в предыдущей задаче. Часть модели, реализующая продвижение модельного времени по особым состояниям, представлена на рис. 3.4.

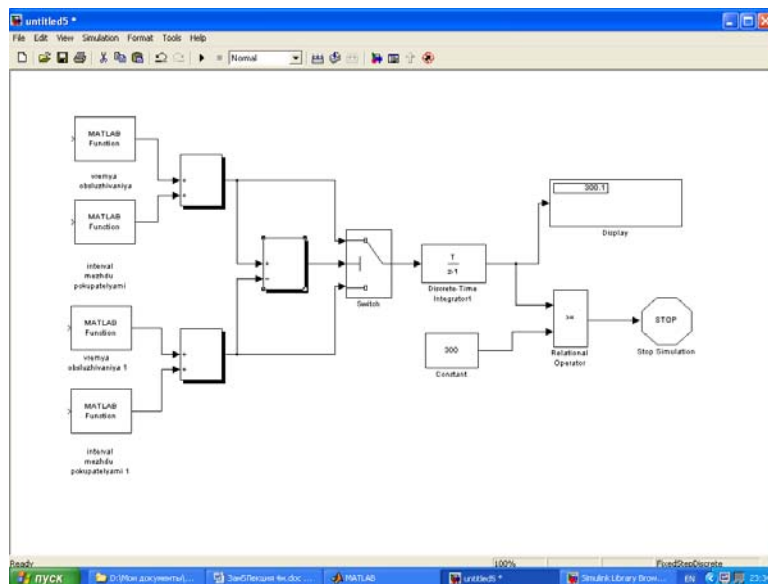


Рис. 3.4. Часть модели двухканальной системы массового обслуживания

2. Поясните назначение блоков сумматоров 1,2,3 и блока Switch Discrete Time Integrator в модели.
3. Проведите моделирование. Опишите результаты моделирования.

Практическое занятие 4

Имитационная модель циклов роста и падений в экономике (кризисов)

Цель работы:

Используя метод имитационного моделирования, исследовать причинно-следственный механизм возникновения циклов и кризисов перепроизводства (на примере автомобильной промышленности).

1. Используя блок-схему модели, приведенную в лекции, построить имитационную модель кризисов, дополнив ее имитацией случайных факторов.

Таковыми факторами могут быть:

- растущий, но ежегодно колеблющийся спрос на продукцию;
- срок службы товара как случайная величина.

Вид закона распределения случайного фактора и его параметры задайте самостоятельно и поясните, почему вы приняли именно такой закон и параметры.

2. Задайте параметры блоков модели, придав параметрам конкретное физическое толкование. Добейтесь работоспособности модели путем подбора параметров и пробных прогонов модели.
3. Поясните в отчете назначение блоков модели и параметров блоков модели.
4. Исследуйте с помощью построенной модели зависимость устойчивости системы (производства авто) при различных лагах производства. Результаты с пояснениями приведите в отчетах.

5. Что происходит при увеличении задержки производства, т.е. отставания реакции производства на спрос, почему?
6. Исследуйте влияние параметра срока службы на показатели экономической системы (устойчивость, появление кризисов). Приведите результаты в отчетах, поясните их.
7. Основываясь на результатах моделирования, укажите, при каких параметрах модели система будет устойчива?
8. Учитывая, что под начальным дефицитом понимается разница между необходимым и реальным парком автомобилей на момент моделирования, измените величину начальных условий на интеграторе блока «Поступление» и исследуйте влияние дефицита на показатели экономической системы, т.е. устойчивость, возможность кризисов.
9. Что, на ваш взгляд, упрощенно моделируется в данной модели? Что надо изменить в модели, чтобы повысить адекватность моделирования?
10. Придайте случайным факторам конкретное экономическое или техническое толкование.

Практическое занятие 5

Построение имитационной модели для определения оптимальной ставки налогообложения прибыли предприятия

Цель работы:

1. Построение модели.
2. Написание программы сценария проведения двухфакторного эксперимента.
3. Исследование зависимости поступлений в бюджет от величины налоговой ставки.

Порядок выполнения работы

1. Используя блок-схему, приведенную в лекции, построить имитационную модель для определения оптималь-

ной ставки налогообложения прибыли предприятия.

2. Поясните в отчете назначение блоков модели и параметров блоков модели.
3. Провести однофакторный (не автоматизированный) эксперимент с построенной моделью. Исследовать зависимость налоговых поступлений за конкретный период времени от величины налоговой ставки на прибыль предприятия.

Запустив модель из меню Simulation, наблюдать в окнах Score изменения показателей предприятия и бюджета во времени: рост поступлений прибыли, отчислений по налогу в бюджет и капитализацию нераспределенной прибыли. Устанавливая различные ставки налога, выполнить прогон модели. Провести анализ экспериментальных графиков накопления средств в бюджет за время моделирования и графиков изменения капитала. В отчете объяснить поведение поступления в бюджет при изменении ставки налогообложения.

4. Используя инструментарий Simulink, написать программу сценария для проведения двухфакторного эксперимента (m-файл по образцу, приведенному в лекции). Написанный сценарий привести в отчете. Пояснить команду plot.
5. Используя построенную имитационную модель, исследовать зависимость бюджетно-оптимальной ставки от эффективности работы фирмы. В качестве показателя эффективности выбрать рентабельность, т.е. отношение доналоговой прибыли к капиталу. Проведением эксперимента управлять командой Tools-Run из m-файла. Провести двухфакторный имитационный эксперимент в автоматизированном режиме. Добиться построения графиков зависимостей поступлений в бюджет от ставок налогов и рентабельности. Построенный график привести в отчете с пояснениями поведения полученных зависимостей, отвечая на вопросы:

Что происходит с графиком зависимости поступлений в бюджет при изменении рентабельности предприятия? Почему?

Используя результаты моделирования, сформулируйте, предприятия с какой рентабельностью следует облагать высокими налогами и почему?

Практическое занятие 6

Построение «паутинообразной» модели фирмы.
(Модель равновесия на конкурентном рынке)

Цель работы:

1. Построение модели.
2. Написание программы сценария автоматизации управления экспериментом.
3. Исследование переходного процесса к рыночному равновесию:
 - исследование влияния смещения линий спроса и предложения на рыночное равновесие;
 - исследование влияния крутизны линии спроса и предложения на рыночное равновесие.

Порядок выполнения работы

1. Используя блок-схему, приведенную в лекции, построить имитационную модель равновесия на конкурентном рынке.
2. Поясните в отчете назначение блоков модели и параметров блоков модели.
3. Используя инструментарий Simulink, написать программу сценария для автоматизации управления имитационным экспериментом по аналогии с предыдущей практической работой.
 - `open_system("...")`
 - `sim("...")`
 - `plot (ScopeData(:,2),ScopeDate(:,3:4))`
 - `hold on`
 - `grid`

- pause(5)
- sim("...")
- for i=2:11

```
line([ScopeDate(i-1,2) ScopeDate(i,2)], [ScopeDate(i,4) ScopeDate (i,4)])  
line([ScopeDate(i,2) ScopeDate(i,2)], [ScopeDate(i,4) ScopeDate (i+1,4)])
```

- end
- hold off

Написанный сценарий привести в отчете. Пояснить все строки программы.

4. Запуская m-файл, начать моделирование в первом режиме моделирования (ключи находятся в нижнем положении). Пояснить графики, наблюдаемые в окнах Scope. Перевести ключи в нижнее положение (на это отводится 5 секунд), задав тем самым второй режим моделирования, наблюдать переходной процесс рынка в равновесное состояние. Построенные графики привести в отчете.

Контрольные работы (для заочного отделения)

1. Моделирование случайных событий и величин (глава 3).
2. Разработка модели потока заявок на обслуживание (глава 5).
3. Имитационное моделирование циклов роста и падений в экономике (кризисов) (глава 9).
4. Имитационная модель для определения оптимальной ставки налогообложения прибыли предприятия (глава 9).
5. «Паутинообразная» модель фирмы (модель равновесия на конкурентном рынке) (глава 9).

Темы лабораторных (семестровых) работ

1. Основы работы в MATLAB/SIMULINK (глава 4).
2. Моделирование случайных событий и величин (глава 3).
3. Управление модельным временем. Моделирование асинхронных процессов (глава 5).
4. Имитационная модель циклов роста и падений в экономике (кризисов) (глава 9).
5. Построение имитационной модели для определения оптимальной ставки налогообложения прибыли предприятия (глава 9).
6. Построение «паутинообразной» модели фирмы (Модель равновесия на конкурентном рынке) (глава 9).

Итоговые вопросы

1. Основы принятия решений относительно создания, совершенствования, развития экономических систем.
2. Основы имитационного моделирования. Понятие модели. Классификация моделей.
3. Основы имитационного моделирования. Последовательность разработки математических моделей.
4. Классификация моделей.
5. Метод Монте-Карло.
6. Классификация моделируемых систем.
7. Математические схемы (модели).
8. Моделирование случайных событий. Моделирование простого события.
9. Моделирование случайных событий. Моделирование полной группы несовместных событий.
10. Моделирование случайных величин. Моделирование дискретной случайной величины.
11. Моделирование случайных величин. Моделирование непрерывных случайных величин. Моделирование случайных величин с заданными параметрами средствами MATLAB.
12. Библиотека блоков Simulink. Источники сигналов.
13. Библиотека блоков Simulink. Приемники сигналов.
14. Библиотека блоков Simulink. Аналоговые блоки.
15. Библиотека блоков Simulink. Нелинейные блоки.
16. Библиотека блоков Simulink. Блоки преобразования сигналов и вспомогательные блоки.
17. Библиотека блоков Simulink. Блоки функций и таблиц.
18. Библиотека блоков Simulink. Команды построения графиков.
19. Управление модельным временем. Виды представления времени в модели. Изменение времени с постоянным шагом.
20. Управление модельным временем. Виды представления времени в модели. Продвижение времени по особым состояниям.
21. Управление модельным временем в MATLAB.

22. Управление модельным временем в MATLAB. Синхронизация параллельных процессов.
23. Установка параметров вывода выходных сигналов моделируемой системы `output options` (параметры вывода). Установка параметров обмена с рабочей областью.
24. Классификация моделей экономических систем. Характеристика общих экономических моделей.
25. Классификация моделей экономических систем. Макроэкономические модели.
26. Классификация моделей экономических систем. Модели управления предприятием.
27. Процессы массового обслуживания в экономических системах; Потоки, задержки обслуживание.
28. Моделирование процессов обслуживания заявок в условиях отказов.
29. Планирование модельных экспериментов. Цели планирования экспериментов.
30. Стратегическое планирование имитационного эксперимента.
31. Тактическое планирование эксперимента.
32. Возможности Matlab/Simulink по планированию и реализации модельных экспериментов.
33. Примеры построения имитационных моделей. Имитационная модель циклов роста и падений в экономике (кризисов). Постановка задачи на моделирование. Построение концептуальной модели. Математическая модель.
34. Использование имитационного моделирования для поиска оптимальной ставки налогообложения на прибыль. Постановка задачи на моделирование. Построение концептуальной модели. Математическая модель. Компьютерная модель в программе Simulink. Исходные данные для параметров, переменных и показателей модели. Математическая схема модели и метод решения. Средства управления экспериментом. Программа управления имитационным экспериментом.
35. «Паутинообразная» модель фирмы (равновесие на конкурентном рынке). Постановка задачи на моделирование. Построение модели.

Глоссарий

- Адекватность модели* – степень соответствия моделируемого процесса процессу функционирования реальной системы.
- Активный имитационный эксперимент* – эксперимент, в котором исследователь может изменять уровни факторов.
- Аналитическое моделирование* – описание процессов функционирования системы с помощью системы алгебраических, дифференциальных, интегральных или конечно-разностных уравнений.
- Вторичные факторы* – факторы, которые не являются предметом исследования, но влиянием которых нельзя пренебречь.
- Гомоморфность модели* – отражение в модели не всех, а только существенных свойств объекта-оригинала.
- Датчик (генератор) случайных чисел* – специальная программа, входящая в состав программного обеспечения ЭВМ, реализующая бросание жребия.
- Движение системы* – процесс смены состояний системы.
- Детерминированная система* – система, в которой новое состояние зависит только от времени и текущего состояния системы.

<i>Динамическая система</i>	- система, в которой множество состояний больше одного и они могут изменяться во времени.
<i>Дискретная случайная величина (ДСВ)</i>	- величина, принимающая конечное (счетное) множество возможных значений.
<i>Дискретно-детерминированная модель</i>	- модель, построенная на основе так называемых конечных автоматов. Автомат можно представить как некоторое устройство, на которое подаются входные сигналы и снимаются выходные и которое может иметь некоторые внутренние состояния.
<i>Дискретно-стохастическая модель</i>	- модель, построенная на основе вероятностных (стохастических) автоматов.
<i>Дробный факторный эксперимент (ДФЭ)</i>	- эксперимент, в котором каждый фактор варьируется на двух уровнях – нижнем и верхнем.
<i>Идентификация факторов</i>	- процесс ранжирования факторов по степени влияния на значение наблюдаемой переменной (показателя эффективности).
<i>Имитационное моделирование</i>	- описание процесса функционирования системы во времени, причем имитируя элементарные явления, составляющие процесс, с сохранением их логической структуры и последовательности протекания во времени.

- | | |
|---|---|
| <i>Имитационный эксперимент</i> | - наблюдение за поведением экономической системы в течение некоторого промежутка времени с помощью модели. |
| <i>Имитация по особым состояниям</i> | - системное время каждый раз изменяется на величину, строго соответствующую интервалу времени до момента наступления очередного события. |
| <i>Имитация с постоянным шагом</i> | - отсчет системного времени ведется через фиксированные, выбранные исследователем интервалы времени. |
| <i>Инициализация модели</i> | - подключение библиотечных блоков к модели, определение размерностей сигналов, типов данных, величин шагов модельного времени, оценка параметров блоков, а также определение порядка выполнения блоков и выделение памяти для проведения расчета. |
| <i>Интервал варьирования фактора</i> | - некоторое число, прибавление которого к нулевому уровню дает верхний уровень, а вычитание – нижний. |
| <i>Информационная достаточность</i> | - достаточное представление разработчика о том, что является входными и выходными переменными в исследуемой системе и какие факторы оказывают влияние на процесс ее функционирования. |

<i>Комбинированное моделирование</i>	- сочетание аналитического и имитационного моделирования.
<i>Концептуальная (содержательная) модель</i>	- абстрактная модель, определяющая структуру моделируемой системы, свойства ее элементов и причинно-следственные связи, присущие системе и существенные для достижения цели моделирования.
<i>Критерий эффективности</i>	- правило, на основании которого производится выбор стратегии, отвечающей интересам ЛПР.
<i>Латинский план</i> (или «латинский квадрат»)	план проведения эксперимента с одним первичным фактором и несколькими вторичными.
<i>ЛПР</i>	- лицо, принимающее решение.
<i>Макроэкономические модели</i>	- модели, имитирующие экономические системы крупного масштаба, такие как область или страна в целом.
<i>Математическая схема</i>	- основа разрабатываемого конкретного моделирующего алгоритма.
<i>Машинное время</i>	- время, отражающее затраты времени ЭВМ на проведение имитации.
<i>Метод Монте-Карло</i>	- численный метод проведения аналитических расчетов с помощью датчиков случайных чисел (получил название метода статистических испытаний).

- Модели систем массового обслуживания*
- модели, предназначенные для установления зависимостей между характером потока заявок, числом каналов обслуживания, производительностью отдельного канала и эффективным обслуживанием с целью нахождения наилучших путей управления этими процессами.
- Модели торговли*
- (модели фирмы) – модели, предназначенные для прогнозирования условий, при которых пункты торговли могут считаться эквивалентными по получаемой прибыли.
- Модели управления запасами*
- модели, предназначенные для прогнозирования суммарных расходов на содержание склада.
- Модели управления предприятием*
- микроэкономические модели, отличающиеся друг от друга не столько областью применения, сколько тем, какая типовая математическая схема заложена в основу модели и каковы особенности используемого математического аппарата.
- Модель*
- объект любой природы, который создается исследователем с целью получения новых знаний об объекте-оригинале и отражает только существенные (с точки зрения разработчика) свойства оригинала.

<i>Модельное время</i> (системное)	- время, в масштабе которого организуется работа модели.
<i>Независимый параллельный процесс</i> (ПП)	- процесс, который не является подчиненным ни для одного из процессов.
<i>Непрерывная случайная величина</i> (НСВ)	- величина, принимающая любые значения из некоторого интервала.
<i>Непрерывно-детерминированная модель</i>	- модель, описываемая системами обыкновенных дифференциальных уравнений или уравнений в частных производных.
<i>Отказы второго рода</i> (аварии)	- отказы, приводящие к такому состоянию системы, что после устранения отказа процесс обслуживания заявки начинается сначала.
<i>Отказы первого рода</i> (неисправности)	- отказы, приводящие к временному прекращению процесса обслуживания очередной заявки с сохранением достигнутого состояния. После устранения отказа процесс обслуживания заявки может продолжаться.
<i>Отраслевые модели</i>	- комплексные, или агрегированные, модели, описывающие отдельные отрасли народного хозяйства как единое целое.

<i>Параллельные процессы (ПП)</i>	- процессы, протекающие в системе одновременно.
<i>Пассивный имитационный эксперимент</i>	- эксперимент, в котором исследователь не может изменять уровни факторов.
<i>Первичные факторы</i>	- те факторы, в исследовании влияния которых экспериментатор заинтересован непосредственно.
<i>Плотность вероятностей (плотность распределения)</i>	- производная функция распределения, характеризующая плотность, с которой распределяются значения случайной величины в данной точке.
<i>Показатель эффективности</i>	- комплексное операционное свойство (качество) процесса функционирования системы, характеризующее его приспособленность к достижению цели операции (выполнению задачи системы).
<i>Полная группа несовместных событий (ПГНС)</i>	- события, сумма вероятностей появления которых равна 1.
<i>Полный факторный эксперимент (ПФЭ)</i>	- эксперимент, в котором реализуются всевозможные сочетания уровней факторов.
<i>Поток событий</i>	- совокупность заявок в системе массового обслуживания.

<i>Производственные модели</i>	- имитационные модели нескольких цехов, которые последовательно участвуют в процессе производства некоторого изделия. Предназначены для оценивания и прогнозирования прибыли. Представляют собой модели систем массового обслуживания без отказов с неограниченным ожиданием.
<i>Рабочая нагрузка</i>	- совокупность внешних воздействий, оказывающих влияние на эффективность применения данной системы в рамках проводимой операции.
<i>Рандомизированный план</i>	- предполагает выбор сочетания уровней для каждого прогона имитационной модели случайным образом. При использовании этого метода отправной точкой в формировании плана является число экспериментов, которые считает возможным (или необходимым) провести исследователь.
<i>Реальное время</i>	- время, в котором происходит функционирование имитируемой системы.
<i>Синхронный параллельный процесс (ПП)</i>	- такой процесс, состояние которого зависит от состояния взаимодействующих с ним ПП.

- | | |
|--|---|
| <i>Системы с дискретными состояниями</i> | - системы, в которых в любой момент времени можно однозначно определить, в каком именно состоянии находится система. |
| <i>Случайная величина (СВ)</i> | - величина, которая в результате опыта может принимать некоторое неизвестное заранее значение. |
| <i>СМО</i> | - система массового обслуживания. |
| <i>Список будущих событий</i> | - список, содержащий события, время наступления которых больше текущего модельного времени, то есть события, которые должны произойти в будущем (условия наступления которых уже определены). |
| <i>Список прерываний</i> | - список, содержащий события, связанные с возобновлением обработки прерванных транзактов. События из этого списка выбираются в том случае, если сняты условия прерывания. |
| <i>Список текущих событий</i> | - список, в котором находятся события, время наступления которых меньше или равно текущему модельному времени. |
| <i>Статическая система</i> | - система, в которой множество ее состояний содержит один элемент. |
| <i>Стохастическая система</i> | - система, в которой можно указать лишь множество возможных состояний перехода и, в некоторых случаях, - вероятностные характеристики перехода в каждое из этих состояний. |

<i>Стратегическое планирование имитационного эксперимента</i>	- выбор из всех допустимых планов такого, который позволил бы получить наиболее достоверное значение функции отклика $f(x)$ при фиксированном числе опытов. Другими словами, стратегическое планирование позволяет ответить на вопрос, при каком сочетании уровней внешних и внутренних факторов может быть получена наиболее полная и достоверная информация о поведении системы.
<i>Субъективность модели</i>	- отражение индивидуальности исследователя при разработке модели.
<i>Тактическое планирование имитационного эксперимента</i>	- выбор из всех допустимых планов такого, который позволил бы получить статистическую оценку функции отклика с заданной точностью при минимальном объеме испытаний. Другими словами, это совокупность методов установления необходимого объема испытаний.
<i>Транзакт</i>	- динамический объект имитационной модели, представляющий формальный запрос на какое-либо обслуживание (заявка (событие)).
<i>Уровни фактора</i>	- значения факторов.

- Файл-сценарий (Script files, M-сценарий)** – последовательность команд (или операторов Matlab), разделенных точкой с запятой и обеспечивающих автоматизированный прогон модели при различных значениях параметров (факторов).
- Факторное пространство** – множество внешних и внутренних параметров модели, значения которых исследователь может контролировать в ходе подготовки и проведения модельного эксперимента.
- Финансовая модель** – модель, предназначенная для прогнозирования объема капиталовложений в условиях неопределенности.
- Функция отклика** – результат работы имитационной модели.
- Функция распределения (интегральная функция распределения, или интегральный закон распределения)** – функция, полностью характеризующая случайную величину с вероятностной точки зрения, т.е. является одной из форм закона распределения.
- Центр плана эксперимента** – точка в факторном пространстве, соответствующая нулевым уровням всех факторов.
- Цикл моделирования** – временной шаг работы модели, на котором происходит расчет блоков в порядке, определенном на этапе инициализации.

<i>Частичный факторный эксперимент (ЧФЭ)</i>	- эксперимент, в котором отсутствует исследование взаимного влияния факторов.
<i>Command History (история команд)</i>	- окно протокола вашей работы.
<i>Command Windows</i>	- окно для ввода и исполнения команд.
<i>Current Direktoxy</i>	- текущий, рабочий справочник (папка).
<i>M-файлы (с расширением .m)</i>	- файлы, содержащие тексты программ на языке MATLAB.
<i>MAT-файлы (с расширением .mat)</i>	- файлы данных, хранящиеся в рабочем пространстве (Workspace) MATLAB.
<i>Mdl-файлы (с расширением .mdl)</i>	- файлы моделей SIMULINK.
<i>Simulink</i>	- программа, являющаяся приложением к пакету MATLAB, предназначенная для имитационного моделирования. В ней реализуется принцип визуального программирования, в соответствии с которым пользователь на экране из библиотеки стандартных блоков создает модель устройства, процесса или системы и осуществляет расчеты.

- Simulink Library Browser*** - окно обозревателя библиотеки блоков Simulink.
- Workspace*** - рабочее пространство MATLAB. В этой оперативной памяти сохраняются все данные рабочей сессии MATLAB.

Список рекомендуемой литературы

1. Анфилатов В.С., Емельянов А.А., Кукушкин А.А. Системный анализ в управлении. – М.: Финансы и статистика, 2001. – 368 с.
2. Борщев А.В. Практическое агентное моделирование и его место в арсенале аналитика // Exponenta Pro. – 2004. – № 3–4 (<http://www.gpss.ru/index-h.html>).
3. Бусленко Н.П. Моделирование сложных систем. – М.: Наука, 1978. – 399 с.
4. Варфоломеев В.И. Алгоритмическое моделирование элементов экономических систем. – М.: Финансы и статистика, 2000. – 208 с.
5. Гультияев А. Визуальное моделирование в среде MATLAB. – СПб.: Питер, – 2000.
6. Гультияев А.К. MATLAB 5.3 Имитационное моделирование в среде Windows: Практик. пособие. – СПб.: КОРОНА принт, 2001. – 400 с.
7. Дьяконов В.П. MATLAB 6.5 SP1/7+ Simulink 5/6. Основы применения. – М.: СОЛОН-Пресс, 2005. – 800 с. (Серия «Библиотека профессионала»).
8. Дэбни Дж.Б., Харман Т.Л. Секреты мастерства. /Пер. с англ. М.Л. Симонова. Simulink 3. – М.: БИНОМ, 2003. – 403 с.
9. Емельянов А.А. Имитационное моделирование в управлении рисками. – СПб.: Инжекон, 2000. – 376 с.
10. Емельянов А.А., Власова Е.А. Имитационное моделирование в экономических информационных системах. – М.: Изд-во МЭСИ, 1998. – 108 с.
11. Емельянов А.А., Власова Е.А., Дума Р.В. Моделирование экономических процессов. – М.: Финансы и статистика, 2005. – 368 с.
12. Клейнрок Л.Т. Теория массового обслуживания. – М.: Машиностроение, 1979. – 432 с.
13. Кобелев Н.Б. Практика применения экономико-математических методов и моделей. – М.: ЗАО «Финстатинформ», 2000.

14. Лоу А.М., Кельтон В.Д. Имитационное моделирование. – СПб.: Питер; Киев: ВНУ, 2004. – 847 с.
15. Налимов В.В., Чернова И.А. Статистические методы планирования экстремальных экспериментов. – М.: Наука, 1965. – 366 с.
16. Нейлор Т. Машинные имитационные эксперименты с моделями экономических систем. – М.: Мир, 1975. – 392 с.
17. Прицкер А. Введение в имитационное моделирование и язык СЛАМ-II. – М.: Мир, 1987. – 544 с.
18. Романцев В.В., Яковлев С.А. Моделирование систем массового обслуживания. – СПб.: Полином, 1995.
19. Рыжиков Ю.И. Имитационное моделирование. Теория и технология. – СПб.: КОРОНА принт, 2004. – 384 с.
20. Саати Т. Элементы теории массового обслуживания и ее приложения. – М.: Сов. радио. – 377 с.
21. Советов Б.Я., Яковлев С.А. Моделирование систем. – М.: Высшая школа, 1999.
22. Фомин Г.П. Системы и модели массового обслуживания в коммерческой деятельности: Учебное пособие. – М.: Финансы и статистика, 2000. – 144 с.
23. Цисарь И.Ф. Лабораторные работы на персональном компьютере: Учебное пособие для студентов экономических специальностей. – М.: Экзамен, 2002. – 224 с.
24. Шеннон Р. Имитационное моделирование систем: искусство и наука. – М.: Мир, 1978.
25. Шмидт Б. Искусство моделирования и имитации. Введение в имитационную систему Simplex 3. Международное общество моделирования и имитации SCS. – Европейское издательство, 2003.
26. Шрайбер Т.Дж. Моделирование на GPSS. – М.: Машиностроение, 1979. – 592 с.
27. Sheldon M. Ross. Simulatoin. – Academic Press. – 3d edition, 2002.