МИНИСТЕРСТВО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ВОЛГОГРАДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ

В. С. Лукьянов, Г. В. Слесарев

ПРОЕКТИРОВАНИЕ КОМПЬЮТЕРНЫХ СЕТЕЙ МЕТОДАМИ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ

Учебное пособие

Волгоград 2001

УДК 62.529

Рецензенты: В. Н Крымов, М.В. Белодедов

Лукьянов В. С., Слесарев Г. В. Проектирование компьютерных сетей методами имитационного моделирования: Учеб. пособие/ ВолгГТУ. - Волгоград, 2001. - 72с.

ISBN 5-230-03878-0

Показана сущность имитационного моделирования, приведены области и основные модели использования средств имитационного эксперимента на ЭВМ. Изложены основные методы формирования дискретных и непрерывных случайных величин в соответствии с заданными законами их распределения. Приведены блок-схемы алгоритмов систем массового обслуживания, для анализа надежности систем.
Пособие предназначено для студентов специальностей 2201, 2202, 2203.

Ил. 32. Табл.2. Библиогр.: 3 назв.

Печатается по решению редакционно-издательского совета Волгоградского государственного технического университета

Оглавление

ВВЕДЕНИЕ

В процессе абстрактного мышления исследователем создаются обобщенные аналогии исследуемого объекта. При этом в аналогии отбрасываются несущественные стороны объекта, не влияющие на результаты исследования. С повышением уровня абстрагирования отбрасывается большее число несущественных сторон объекта. В целом сходство аналогии с объектом уменьшается, но упрощается сама аналогия. Аналогии, отражающие исследуемый объект, должны реализовать в виде удобных для исследования логических схем, которые называются моделями. Итак, модель – это объект, замещающий более сложный объект – оригинал для упрощения его исследования. Соответственно, моделирование есть представление объекта моделью с целью получения информации об этом объекте путем проведения экспериментов.

Модель всегда приближена, упрощена по отношению к объекту. Если результаты моделирования подтверждаются по поведению объекта и могут служить основой для прогнозирования протекающих в нем процессов, то это служит признаком удачной модели, адекватности ее к объекту. Адекватная модель в силу возможностей ее всестороннего исследования позволяет получить новые сведения об объекте.

Настоящее время характеризуется резким возрастанием роли моделирования во всех сферах и отраслях науки и техники. Это обусловлено созданием все более сложных технических систем, требующих комплексного исследования. Особую актуальность моделирование приобретает в области проектирования компьютерных сетей. При проектировании этих систем возникают многочисленные задачи, требующие оценки количественных и качественных закономерностей процессов функционирования этих систем, проведения анализа их работы. Эти системы относятся к сложным системам, характеризующимся сложностью структуры, случайностью (стохастичностью) связей между отдельными их элементами, неоднозначностью поведения при различных внутренних состояниях и внешних воздействиях, большом количестве случайных факторов,

переменчивым характером исходной информации. Недоработки, ошибочные решения, принимаемые на этапе разработки этих систем, приводят к большим экономическим затратам после их изготовления и внедрения на производстве.

С учетом сложности такой системы, основным средством ее исследования на раннем этапе разработки является моделирование. По сути дела моделирование является единственной возможностью создания эффективной системы организации оптимальных режимов ее эксплуатации.

1. СУЩНОСТЬ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ. ОБЛАСТИ ПРИМЕНЕНИЯ ИМИТАЦИОННЫХ МОДЕЛЕЙ

1. 1. Области применения имитационного моделирования в машиностроении

Современный этап развития машиностроения характеризуется бурным внедрением автоматизированных систем управления технологическими процессами и производством в целом, гибких автоматизированных производств на базе робототехнических систем. Разработка, внедрение и организация оптимальных режимов эксплуатации этих сложнейших систем немыслимо без имитационного моделирования.

Для выяснения роли имитационного моделирования в машиностроении рассмотрим простой пример. Допустим, что, в общем, объем деталей, используемых при сборке изделия, половина деталей имеют размеры с положительными отклонениями, а другая половина - с отрицательными отклонениями от номинала. При сборке на изделие устанавливаются пять деталей. В случае попадания всех деталей с одинаковыми положительными или отрицательными отклонениями изделие бракуется. Необходимо определить вероятность брака при условии, что сборщик (например, робот) выбирает деталь наугад. Аналитическая модель здесь проста. Для большого числа исходных деталей вероятность брака Рбр = 2 -5 = 0, 5 5 = 0, 1375.

Решим задачу розыгрышем, путем бросания монеты пять раз подряд. Если во всех случаях при бросании монеты выпадает герб или только плата, будем считать изделие бракованным. Повторим этот опыт много раз. Для общего числа N испытаний получим К случаев попаданий только одной стороны монеты (герба или платы). Тогда отношение К / N будет в определенной степени соответствовать вероятности брака. Результат будет достоверным лишь при большом числе испытаний. Уже на этом примере видно, что для проведения статистического моделирования требуется большое число испытаний, которое без помощи ЭВМ реализовать практически невозможно.

В данном примере аналитическое решение задачи оказалось проще.

Однако при исследовании сложных процессов и систем. подверженных случайным факторам, аналитическое решение может оказаться сложным и даже невозможным.

Рассмотрим более сложную ситуацию. В парк станков поступают разнородные заготовки, требующие разного времени обработки в зависимости от типа изготавливаемой детали, требуемых приспособлений, квалификации рабочего. Поток поступающих заготовок случаен во времени и может изменяться в течение рабочего дня (так называемый случайный поток с изменяющимися параметрами). Отдельные станки могут на время выходить из строя. Неизбежны также случаи прерывания работы, например, для кратковременного отдыха рабочего. Кроме того, одна заготовка может проходить последовательную обработку на разных станках, т. е. в этом случае необходимо найти рациональные режимы группы взаимосвязанных станков. Аналитическое решение по оптимизации работы станочного парка с целью обеспечения максимальной загрузки станков и получения максимальной общей производительности, как правило, в таких ситуациях невозможно. В то же время, как показала практика, имитационное моделирование позволяет справиться с решением данной задачи. Такие задачи относятся к задачам массового обслуживания в машиностроении.

Задачи массового обслуживания в машиностроении весьма разнообразны. Особую актуальность они приобрели в условиях развертывания гибкого автоматизированного производства на базе робототехнических систем. Типичный пример - разработка программы управления роботом при организации многостаночного обслуживания. Необходимо оперативно найти оптимальный вариант обслуживания роботом парка станков, обрабатывающих в той или иной последовательности группу деталей. Аналитические решения в основном применяются лишь в простейших случаях при последовательной обработке одного типа детали при постоянном времени ее обработки на каждом станке /I, 2/.

Выбор типа ЭВМ для управления технологическими процессами в АСУТП и ГАП непосредственно зависит от интенсивности потока заявок, поступающих на нее с объектов управления (станков с числовым программным управлением, автоматизированных рабочих мест, измерительных и контролирующих устройств датчиков и т. п.). В этом случае опять возникает задача массового обслуживания, решаемая имитационным моделированием. Можно привести целый ряд других задач машиностроения, решаемых с помощью имитационного моделирования.

Управление запасами. Решение таких задач связано с определением рациональных объемов запасных частей, материалов, комплектующих элементов, а также времени их смены и пополнения с учетом случайных колебаний времени.

Замена основного и вспомогательного оборудования вследствие морального и физического износа и поломок.

Оптимизация сроков профилактических проверок и ремонтных работ, обеспечивающих максимальный коэффициент их работоспособного состояния.

Оптимальное распределение ресурсов и плановых заданий между подразделениями.

Анализ сложных конфликтных ситуаций в условиях автоматизированного производства в условиях имеющейся неопределенности. В процессе имитационного моделирования находятся компромиссные решения, направленные на уменьшение общих расходов на простом оборудовании, неизбежные при конфликтных ситуациях (например, простой станка из-за отсутствия поступлений из заготовительного цеха и т. п.).

Широко применяется имитационное моделирование при сетевом планировании и управлении. Аналитические методы расчета сетевой модели со случайными временными оценками работ являются либо весьма громоздкими, либо содержат ошибки систематического характера. Обязательным элементом подготовки специалистов по организации

управления и планирования в настоящее время является проведение имитационного моделирования, позволяющего получить временные оценки по отдельным видам работ, предусмотренных в сетевом графике, с требуемой точностью.

Таким образом, на современном этапе машиностроения имитационное моделирование используется практически на всех стадиях и сферах производства. Ниже будут рассмотрены отдельные задачи машиностроения, решаемые методами имитационного моделирования.

1. 2. Виды моделирования систем

Моделирование является одним из основных средств исследования закономерностей объекта, явления, процесса. Известны три основные формы моделей — аналитические, имитационные и экспериментальные.

Аналитическая модель базируется на математическом описании объекта /1,2 /. Анализ характеристик объекта по аналитическим зависимостям может быть проведен лишь при значительной степени абстракции модели по отношению к отображаемому объекту. Как правило, при создании аналитической модели приходится идти на существенные упрощения и допущения, что может привести к получению лишь общих приближенных и даже недостоверных результатов. Если выбранные критерии, характеризующие поведение объекта, удается выразить в виде аналитических зависимостей, то такая модель имеет решение. Однако при исследовании многих объектов аналитическое решение в явном виде не получается. Например, часты ситуации, когда решение представляется в виде преобразования Лапласа, или в виде системы сложных интегро-дифференциальных уравнений. Следует отметить, что в последние годы возможности исследования аналитических моделей значительно возросли благодаря бурному развитию и внедрению методов вычислительной математики с численным решением на ЭВМ.

Важным преимуществом аналитических моделей в целом является возможность быстрого с минимальными затратами получения значений параметров исследуемого объекта. При построении и применении

аналитических моделей важно выбрать лишь существенные параметры и отбросить параметры, мало влияющие на качество функционирования объекта (системы). Для этого необходимо хорошо представлять физическую сущность процессов, поведения объекта, чтобы понять, какие упрощающие предположения и допущения мало скажутся на конечных результатах.

Во многих случаях требуется более детальная информация о поведении объекта, системы. В этом случае используют имитационное моделирование, с помощью которого описывается функционирование системы в виде последовательности операций на ЭВМ. Поведение системы представляется в виде алгоритма, на основе которого разрабатывается программа для ЭВМ.

Сущность имитационного моделирования состоит в том, что процесс имитируется с помощью арифметических и логических операций в последовательности, соответствующей моделируемому процессу.

Важное преимущество имитационной модели по отношению к аналитической заключается в том, что за счет детализации ее можно сделать весьма близкой к моделируемому объекту. Однако такое приближение неизбежно связано с усложнением и большим временем разработки имитационной модели. В результате могут возникать ситуации, когда время разработки имитационной модели бывает сравнимо со временем разработки непосредственно системы и, как следствие этого, отпадает необходимость или значительно снижается актуальность получаемых в результате проведения моделирования результатов. Имитационное моделирование целесообразно поэтому применять в тех случаях, когда задачи оценки исследуемой системы сложны и не поддаются аналитическому решению, а также при исследовании долгосрочных проблем.

С учетом сложности разработки имитационных моделей важно создание базовых программных модулей, описывающих типовые ситуации, возникающие при имитационном моделировании. Процесс составления

агрегатных моделей на основе имеющихся модулей значительно ускоряет процесс имитационного моделирования исследуемой системы. Наличие готовых баз модулей, имитирующих процессы поступления заявок в систему из внешней среды (заказов на обслуживание), процессы обработки этих заказов, выдачи результатов и т. п., позволит значительно расширить области применения имитационных моделей в задачах исследования сложных систем.

Экспериментальные модели исторически использовались одними из первых при проведении испытаний, исследовании сложных систем. Они дают наиболее полную и достоверную информацию об исследуемом объекте. В ряде отраслей экспериментальное моделирование является доминирующим при разработке объекта. Например, обязательным этапом разработки новой конструкции самолета является проведение испытаний его модели в аэродинамической трубе. По существу, целые этапы разработки многих видов аппаратуры, комплексов, систем имеют своей конечной целью проведение экспериментов, испытаний разработок при соответствующих условиях эксплуатации. Лишь на основе практики, проведения испытаний можно окончательно судить о качестве разработанного объекта.

В целом, все перечисленные модели, как правило, используются на разных этапах разработки изделий.

Аналитические модели в основном используются на первом этапе для получения общих ориентировочных оценок, помогающих обоснованно выбрать основные принципы и структуру построения проектируемого изделия. С целью более детальной проработки структуры построения отдельных узлов, связей между ними, взаимодействия с внешней средой используется имитационное моделирование. Прогнозируемые на основе аналитического и имитационного моделирования параметры и качество функционирования изделия проверяются на экспериментальной уменьшенной (упрощенной), модели или натурном лабораторном макете. Комплексные результаты исследований учитываются на дальнейшем этапе

разработки опытных образцов и промышленного освоения выпуска изделий.

1. 3. Сущность имитационного моделирования

Имитационное моделирование позволяет воспроизводить процесс функционирования системы во времени с сохранением элементарных явлений, их логической структуры и последовательности протекания во времени. Это позволяет по исходным данным получить сведения о состояниях процесса в будущем в определенные моменты времени. В настоящее время имитационный метод является наиболее эффективным, а нередко и единственным методом исследования сложных систем на этапе их проектирования.

Имитационная модель является динамической моделью, в которой все процессы рассматриваются в неубывающем масштабе времени. Если два явления в моделируемой системе происходят в определенном порядке, то явления, соответствующие или в моделирующей системе, не могут произойти в обратном порядке.

В общем случае имитационная модель может быть реализована разными средствами и способами. Например, известны имитаторы, основанные на гидродинамических, механических, электронных системах. В настоящее время все больше и больше имитационных моделей реализуется на ЭВМ. Основной подход для создания имитационной модели на ЭВМ заключается в формировании на ЭВМ случайных величин и функций и многократного их воспроизведения в соответствии с закономерностями моделируемого процесса. В результате последующей статистической обработки получаемых частных результатов получаются итоговые результаты, характеризующие процесс функционирования системы. Машинный вариант имитационного моделирования называется методом статистического моделирования.

В упрощенном виде метод статистического моделирования был известен задолго до появления ЭВМ как метод Монте-Карло, основанный на ручном получении случайных чисел, наподобие рулетки игрового

автомата.

До появления ЭВМ этот метод не мог найти широкого применения, так как моделирование случайных процессов вручную требовало много времени. Лишь с появлением ЭВМ началось бурное внедрение статистического моделирования во все сферы и отрасли науки и производства как для решения вероятностных, так и детерминированных задач.

1.4. Основные модели имитируемых систем

Выбор имитационной модели определяется видом и характером решаемой задачи. Каждая конкретная система характеризуется набором свойств, отражающих ее поведение и условия функционирования во взаимодействии с внешней средой.

Рассматривая совокупность систем, встречающихся в сфере машиностроения, можно однозначно утверждать, что там встречаются самые разнообразные задачи, требующие для своего решения весьма разных имитационных моделей.

Рассмотрим основные типы имитационных моделей применительно к их использованию для решения задач машиностроения.

В целом имитационные модели делятся на две группы − детерминированные и стохастические. В детерминированной модели как внешние воздействия, так и внутренние параметры строго регламентированы во времени, элементы случайности в них исключаются. В стохастических моделях или внешнее воздействие, или внутренние параметры связаны со случайными факторами. Эти модели оперируют соответственно с дискретным и непрерывным временем.

Непрерывно-детерминированные модели (Д-схемы)

Применяются для исследования систем, функционирующих в непрерывном времени. Для описания таких систем в основном используются дифференциальные, интегральные, интегро-дифференциальные уравнения. В обыкновенных дифференциальных

уравнениях рассматривается функция только одной независимой переменной, а в уравнениях в частных производных − функции нескольких переменных.

В качестве примера применения Д-моделей можно привести исследование работы механического маятника или электрического колебательного контура. Техническую основу Д-моделей составляют аналоговые вычислительные машины (АВМ) или бурно развивающиеся в настоящее время гибридные вычислительные машины (ГВМ). Как известно, основной принцип исследований на ЭВМ состоит в том, что по заданным уравнениям исследователь (пользователь АВМ) собирает схему из отдельных типовых узлов — операционных усилителей с включением цепей масштабирования, демпфирования, аппроксимации и т. п.

Структура АВМ изменяется в соответствии с видом воспроизводимых уравнений.

В цифровой ЭВМ структура остается неизменной, а изменяется последовательность работы ее узлов в соответствии с заложенной в нее программой. Сравнение АВМ и ЦВМ наглядно показывает разницу между имитационным и статистическим моделированием.

АВМ реализует имитационную модель, но, как правило, не использует принципы статистического моделировании. В ЦВМ большинство имитационных моделей базируется на исследовании случайных чисел, процессов, т. е. на статистическом моделировании. Непрерывно-детерминированные модели широко используются в машиностроении при исследовании систем автоматического управления, выборе амортизирующих систем, выявлении резонансных явлений и колебаний в технике
и т. п.

Дискретно-детерминированные модели (F-схемы)

Оперируют с дискретным временем. Эти модели являются основой для исследования работы чрезвычайно важного и распространенного сегодня класса систем дискретных автоматов. С целью их исследования

разработан самостоятельный математический аппарат теории автоматов. На основе этой теории система рассматривается как автомат, перерабатывающий дискретную информацию и меняющий, в зависимости от результатов ее переработки, свои внутренние состояния.

На этой модели основаны принципы минимизации числа элементов и узлов в схеме, устройстве, оптимизация устройства в целом и последовательности работы его узлов. Наряду с электронными схемами, ярким представителем автоматов, описываемых данной моделью, является робот, управляющий (по заданной программе) технологическими процессами в заданной детерминированной последовательности.

Станок с числовым программным управлением также описывается данной моделью. Выбор последовательности обработки деталей на этом станке осуществляется настройкой узла управления (контроллера), вырабатывающего сигналы управления в определенные моменты времени / 4 /.

Теория автоматов использует математический аппарат булевых функций, оперирующих с двумя возможными значениями сигналов 0 и 1.

Автоматы разделяются на автоматы без памяти, автоматы с памятью. Описание их работы производится с помощью таблиц, матриц, графов, отображающих переходы автомата из одного состояния в другое. Аналитические оценки при любом виде описания работы автомата весьма громоздки и уже при сравнительно небольшом числе элементов, узлов, образующих устройство, практически невыполнимы. Поэтому исследование сложных схем автоматов, к которым, несомненно, относятся и робототехнические устройства, производится с применением имитационного моделирования.

Дискретно-стохастические модели (Р-схемы)

Применяются при исследовании работы вероятностных автоматов. В автоматах этого типа переходы из одного состояния в другое осуществляются под воздействием внешних сигналов и с учетом внутреннего состояния автомата. Однако в отличие от Г-автоматов, эти

перехода не строго детерминированы, а могут осуществляться с определенными вероятностями.

Пример такой модели представляет дискретная марковская цепь с конечным множеством состояний. Анализ F-схем основан на обработке и преобразовании матриц вероятностей переходов и анализе вероятностных графов. Уже для анализа сравнительно простых устройств, поведение которых описывается F-схемами, целесообразно применение имитационного моделирования. Пример такого моделирования приведен в пункте 2.4.

Непрерывно-стохастические модели (Q-схемы)

Используются при анализе широкого класса систем, рассматриваемых как системы массового обслуживания. В качестве процесса обслуживания могут быть представлены различные по своей физической природе процессы: потоки поставок продукции предприятию, потоки комплектующих заказных деталей и изделий, потоки деталей на сборочном конвейере, потоки управляющих воздействий от центра управления АСУ на рабочие места и обратные заявки на обработку информации в ЭВМ и т. д.

Как правило, эти потоки зависят от многих факторов и конкретных ситуаций. Поэтому в большинстве случаев эти потоки случайны во времени с возможностью изменений в любые моменты. Анализ таких схем производится на основе математического аппарата теории массового обслуживания. К ним относится непрерывная марковская цепь. Несмотря на значительные успехи, достигнутые в разработке аналитических методов, теория массового обслуживания, анализ Q-схем аналитическими методами может быть проведен лишь при значительных упрощающих допущениях и предположениях. Детальное исследование большинства этих схем, тем более таких сложных, как АСУТП, робототехнические системы, может быть проведено только с помощью имитационного моделирования.

Обобщенные модели (А-схемы)

Основаны на описании процессов функционирования любых систем на базе агрегативного метода. При агрегативном описании система разбивается на отдельные подсистемы, которые могут считаться удобными для математического описания. В результате такого разбиения (декомпозиции) сложная система представляется в виде многоуровневой системы, отдельные уровни (агрегаты) которой поддаются анализу. На основе анализа отдельных агрегатов и с учетом законов взаимосвязей этих агрегатов удается провести комплексное исследование всей системы.

2. ОРГАНИЗАЦИЯ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ НА ЭВМ

2. 1. Способы формирования случайных равномерно распределенных чисел

Как следует из описания сущности имитационного моделирования, основой его является учет случайных воздействий на рассматриваемую систему. Наличие простых и экономичных способов формирования случайных чисел в ЭВМ во многом определяет эффективность и возможности использования данного вида моделирования. Известны три основных способа формирования случайных чисел: табличный, аппаратный, алгоритмический.

При табличном способе случайные числа заранее формируются с помощью специального механического (простейший вид − рулетка) или электронного устройства. Последовательность таких чисел затем записывается в память ЭВМ. Данный способ не получил распространения вследствие трудоемкости заготовки таблиц, необходимости в большом объеме памяти, отводимой под эти таблицы, больших затратах времени в процессе моделирования при размещении этой таблицы во внешнем накопителе.

При аппаратном способе последовательность случайных чисел вырабатывается отдельным электронным узлом-генератором случайных чисел (ГСЧ).

В ГСЧ используется задающий элемент — источник шумов, которые затем усиливаются, селектируются на определенном уровне с последующим формированием последовательности случайных импульсов с равновероятным появлением токовых и бестоковых, или нулевых и единичных импульсов ($P_0 = P_1 = 0,5$) на любом месте такой случайной последовательности. Недостатками данного метода являются необходимость аппаратного усложнения ЭВМ и невозможность повторного получения идентичных последователей, что важно при отладке программ и проведении сравнительных расчетов.

Для обеспечения возможности неоднократного воспроизведения

одинаковых последовательностей можно использовать так называемые генераторы псевдослучайных чисел (ГПСЧ), схемно реализованные в регистрах сдвига с обратными связями. При длине регистра сдвига, равной n разрядов, получается псевдослучайная последовательность с максимальной длиной 2^n-1 импульсов (тактов).

Следует отметить, что с совершенствованием технологии микросхем изготовление большой интегральной схемы (БИС) на одном кристалле, вмещающем в себя регистр с обратной связью, например, на 60-90 разрядов, на сегодняшний день не представляется затруднительным. Естественно, что такой ГПСЧ, имеющий периодичность $2^{90} \approx 10^{27}$ тактов, может рассматриваться как генератор случайных чисел. Например, при моделировании сложной задачи, на которую необходимо затратить восемь часов непрерывной работы шестнадцатиразрядной ЭВМ с расходом случайных чисел через 2 мс потребуется $3600 \cdot 8 \cdot 10^3 \cdot 16 \approx 4{,}5 \cdot 10^8$ тактов. Поэтому наличие в ЭВМ дополнительной БИС, реализующей ГПСЧ, представляется перспективным направлением. Быстродействие ЭВМ при этом выше, чем при использовании других способов получения случайных чисел, так как ЭВМ в процессе моделирования будет использовать готовые числа от внутреннего датчика — БИС ГПСЧ.

На сегодняшний день наибольшее распространение имеет алгоритмический способ получения случайных чисел. Еще на ранних стадиях создания ЭВМ основоположник ЭВМ Нейман предложил следующий способ. Берется произвольное число α_0, состоящее из 2n двоичных цифр. Величина α_0 возводится в квадрат, который состоит уже из 4n цифр. Далее выбирается α_1 из 2n средних двоичных цифр квадрата, и в дальнейшем процесс повторяется в той же последовательности.

В настоящее время в основном используется так называемый конгруэнтный алгоритм Лемера. Числа a и b конгруэнтны по модулю m , если их разность кратна числу m. Отношение сравнения (конгруэнтности) записывают так $a \equiv b \pmod m$. Эта запись означает, что разность $a - b$ делится на m без остатка или, другими словами, числа a и b дают

одинаковый остаток при делении на m. Например, 1364 = 4(mod 10), 1262 = 162(mod 100). Применяются следующие конгруэнтные алгоритмы получения случайных чисел:

$r_{i+1} = a\, r_i \pmod m$ − мультипликативный алгоритм,

где a , m -неотрицательные целые числа,

r_i, r_{i+1} − очередное и последующее случайное число,

$r_{i+1} = ar_i + c \pmod m$ − смешанный алгоритм,

$r_{i+1} \equiv r_i + r_{i-1} \pmod m$ − аддитивный алгоритм.

По данным выражениям легко составляется стандартная подпрограмма получения случайных чисел. Например, при использовании мультипликативного алгоритма для получения последующего случайного числа r_{i+1} нужно взять последнее случайное число r_i умножить его на a и взять модуль полученного числа по m, т.е. разделить на m и принять за остаток r_{i+1}. Выбор a, m, r_0 (первого числа) производится из условия обеспечения минимальной корреляции между генерируемыми числами и максимального периода повторения последовательности цифр. Обычно $m=2^b$, где b − число двоичных цифр в машинном слове.

Во всех выпускаемых в настоящее время ЭВМ, включая микроЭВМ, заложена стандартная подпрограмма *RANDOM* (случайный), сокращенно записываемая *RAND* или *RND,* по которой вырабатывается случайное число, равномерно распределенное в интервале $0 < r_i < 1$. Например, подпрограмма для ЕС ЭВМ, записанная на языке ФОРТРАН, имеет вид

```
SUBROUTINE    RANDU ( IX. , IY , YEL)
IY = IX*— 65539
IF(IY) 5,6,6
5 IY = IY + 2147483647 +1
6 YEL = IY
YEL = YEL*— 4656613E –9
RETURN
END,
```

где IX — число, которое при первом обращении должно содержать

нечетное целое число, состоящее из девяти или менее цифр. После первого обращения IX должно быть равно IY , вычисленному подпрограммой при предыдущем обращении;

IY — полученное целое случайное число, требуемое при последующих обращениях к подпрограмме $0 < IY < 2^{31}$**;**

YEL — полученное равномерно распределенное случайное число $0 < r_i < 1$.

В качестве исходной величины рекомендуется брать число 65539.

2. 2. Моделирование случайных событий и дискретных случайных величин

Как известно из теории вероятностей, вероятность наступления какого-либо события А находится в пределах $0 \leq P(A) \leq 1$. Для моделирования простого события достаточно реализовать в ЭВМ случайное равномерно распределенное число r и считать, что при $r \leq P$ данное событие произошло, а при $r > P$ событие не имеет места (рис. 2. 1). На принципе попадания случайного числа r на то или иное место участка 0-1 можно моделировать более сложные случаи.

Для полной группы событий, имеющих вероятности $P_1+P_2+...+P_n=1$, необходимо интервал 0-1 разделить на n частей, длины каждой из которых равны соответственно значениям P_1, P_2,...P_n. При попадании случайного числа на отдельный участок, например участок К, считается, что наступило событие A_k.

Например, согласно рис. 2.2, наступило событие A_3. При этом $P_2 < r \leq P_3$.

В ходе моделирования генерируемая случайная величина последовательно сравнивается с элементами массива, возрастающих чисел P(i),

где $P(i) = \sum_{j=1}^{i} P(j)$, $P(n) = 1$.

Вариант алгоритма моделирования полной группы событий показан на рис. 2. 3. Значение А будет соответствовать разыгранному событию и используется для дальнейшего решения. Следует заметить, что проверка условия i>n в этом алгоритме является излишней, так как всегда r < 1, а последнее значение элемента массива P (n) =1.

При моделировании комбинаций независимых событий можно применить алгоритм для полной группы событий.

Например, для двух событий А и В возможны следующие варианты АВ, $A\overline{B}$, $\overline{A}B$, $\overline{A}\overline{B}$. При этом всегда P(AB) + P($A\overline{B}$) + P($\overline{A}B$)+ P($\overline{A}\overline{B}$) = 1, где P(AB) = P_a ● P_b, P($A\overline{B}$) = P(A) ● (1 – P_b) и т.д.

На ЭВМ моделируются отдельные события А и В.

При $r_1 < A$ и $r_2 <$ В принимается, что наступило сложное событие АВ;

при $r_1 <$ А, $r_2 >$ В принимается событие $A\overline{B}$;

при $r_1 >$ А, $r_2 <$ В принимается событие $\overline{A}B$;

при $r_1 >$ А, $r_2 >$ В принимается событие $\overline{A}\overline{B}$.

При моделировании сложных зависимых событий А и В, наряду с вероятностями P_a, P_b, необходимо учитывать условную вероятность наступления события В при условии, что событие А имело место ($P_{b/a}$). Эта вероятность задается в качестве исходной величины.

В ходе моделирования могут быть разыграны следующие ситуации:

При $r_1 < P_a$, $r_2 < P_{a/b}$ принимается событие АВ;

$r_1 <$ А, $r_2 > P_{a/b}$ принимается событие $A\overline{B}$;

$r_1 >$ А , $r_2 < P_{b/\bar{a}}$ принимается событие $\overline{A}B$;

$r_1 >$ А , $r_2 > P_{b/\bar{a}}$ принимается событие $\overline{A}\overline{B}$.

Значение $P_{b/a}$ находится по формуле Байеса

$$P_{b/\bar{a}} = \frac{P_b - P_a \cdot P_{b/a}}{1 - P_a}.$$

Целесообразность моделирования простых и сложных событий на ЭВМ, несмотря на сравнительную простоту их аналитического решения, обуславливается тем, что обычно на ЭВМ моделируется более сложная модель, куда процесс получения отдельных групп событий входит составной частью.

Рассмотренный способ моделирования полной группы событий может быть использован также для моделирования дискретной случайной величины X с заданным законом распределения $P(X_i) = P_i$, $i = 1 \div n$,

причем

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

При попадании генерируемого случайного числа r на участок К принимается, что случайная величина X приняла значение X_k.

2. 3. Моделирование случайных величин по заданным законам распределения

Случайные величины характеризуются различными законами распределения. Для одного из них, равномерного, выше были рассмотрены способы генерации случайных чисел. Генерация случайных чисел, подчиняющихся другим законам, основана на базовой модели генерации в ЭВМ равномерно распределенных случайных чисел и последующего преобразования этих чисел. Известны несколько методов преобразования. Ниже рассмотрен один из наиболее распространенных методов преобразования — метод обратной функции.

Как известно, случайная величина X описывается интегральной F(x) и дифференциальной f(x) функциями распределения. Зная одну из этих функций, можно предсказать поведение случайной величины во времени. Обе функции связаны между собой

$$F(x) = \int_{-\infty}^{x} f(x)dx, \qquad f(x) = F'(x).$$

Интегральная функция представляет собой вероятность того, что какое-то взятое фиксированное значение X будет меньше текущего значения x $F(x) = P(X<x)$ (рис. 2.4). Очевидно, что с повышением текущего значения X эта вероятность будет стремиться к 1. Функция F(x) является монотонно возрастающей функцией. Всегда при $x_2 > x_1$ $F(x_2) > F(x_1)$.

Соответственно, при
$P[F(x_1) < F(x_2)] = P(x_1 < x_2)$,

Примем, что случайная величина $r = F(x)$.

Найдем распределение этой величины $F_r(r)$.

На основании приведенных выражений

$F_r(r) = P(R < r) = P[F(X) < F(x)] = F(X < x) = F(x) = r$

$$R = F_r(r) - F(x). \qquad (2.1)$$

Согласно выражению (2.1), вероятность попадания случайной величины в интервал 0 – r равна длине этого интервала, и это есть признак того, что данное распределение равномерное.

В результате получаем алгоритм формирования непрерывной случайной величины X по заданному закону распределения. Поскольку $r_i = F(x_i)$, то необходимо выполнить преобразование

$$X_i = F^{-1}(r_i), \qquad (2.2)$$

где r_i — равномерно распределенное случайное число;

F^{-1} — функция, обратная по отношению к распределению случайной величины X.

На основании выражения (2.2) можно моделировать случайные числа с заданным законом распределения.

2.3.1. Моделирование случайной величины, распределенной по показательному закону

Показательным законом описываются многие физические процессы: случайное время безотказной работы электронных и ряд других изделий, случайные моменты времени поступления заказов на предприятия, службы быта, вывозов на телефонные станции, поступления судов в отдельные порты, времена поиска неисправностей в аппаратуре и

т.д.

Дифференциальная и интегральная функции распределения случайной величины, распределенной по показательному закону, определяются выражениями (рис. 2.5):

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

$F(x) = 1 - e^{-\lambda x}$ $(x>0)$,

где λ — постоянная величина, параметр показательного распределения.

В соответствии с выражением (2.2) имеем $r_i = 1 - e^{-\lambda x_i}$. Разрешив его относительно x_i, получим $x_i = -(1/\lambda)\ln(1-r_i)$.

Поскольку случайное число r_i равномерно распределено в интервале 0-1, то величины $1-r_i$, r_i распределены одинаково. Поэтому для моделирования случайной величины, распределенной по показательному закону, используется выражение

$$x_i = -(1/\lambda)\ln r_i. \qquad (2.3)$$

2.3.2. Моделирование случайной величины, распределенной по линейному закону

Дифференциальная и интегральная функции распределения этого закона выражаются следующим образом (рис. 2.6):

$f(x) = \lambda(1 - \lambda x/2)$,

$F(x) = \lambda x - \lambda^2 x^2/4$, $(0<x<2/\lambda)$.

В соответствии с выражением (2.2) получаем обратное преобразование функции *F (x)*

$$x_i = (2/\lambda)(1 - \sqrt{r_i}). \qquad (2.4)$$

2.3.3. Моделирование случайной величины, распределенной по равномерному закону

Равномерное распределение для интервала (a, b) определяется выражениями (рис. 2.7):

$f(x) = 1/(b-a)$,

$F(X)=(X-A)/(B-A) \qquad a<x \leq b.$

На основании (2.2) имеем

$$x_i=a+(b-a)r_i. \qquad (2.5)$$

2.3.4. Моделирование случайной величины, распределенной по закону Вейбулла

Наряду с показательным распределением, распределение Вейбулла имеет место при оценках модельности изделий. Этому распределению подчиняется случайное время работы изделия до отказа, возникающего вследствие износа и старения.

Дифференциальная и интегральная функции этого распределения имеют вид (рис. 2.8)

$$f(x) = \lambda_0 \alpha x^{\alpha-1} e^{-\lambda_0 x^2},$$

$$F(x) = 1 — e^{-\lambda_0 x^2} \qquad (x>0),$$

где λ_0, α_0 − параметры закона распределения,

$\lambda_x = \lambda_0 \alpha x^{\alpha-1}$— интенсивность отказов.

С учетом выражения (2.2) формула для расчета случайной величины x_i, подчинявшейся распределению Вейбулла, имеет вид

$$x_i = \sqrt[\alpha]{-(1/\lambda_0)\ln r_i}. \qquad (2.6)$$

2.3.5. Моделирование случайной величины, распределенной по нормальному закону

Нормальный закон используется буквально во всех областях, где оперируют со случайными величинами. Этот закон является предельным, к которому приближаются другие законы распределения, описывающие различные процессы.

Функции распределения этого закона имеют вид (рис. 2. 9)

$$f(x) = \frac{1}{\sigma_x \sqrt{2\pi}} \cdot e^{-(x-m_x)^2/(2\sigma_x^2)},$$

$$F(x) = \frac{1}{\sigma_x \sqrt{2\pi}} \cdot \int_{-\infty}^{\infty} e^{-(x-m_x)^2/(2\sigma_x^2)} dx.$$

Непосредственное использование выражения F (x) для расчета случайной величины, распределенной по нормальному закону, требует сравнительно больших затрат машинного времени, так как связано с численным решением не поддающегося аналитическому расчету интеграла. Имеются табулированные значения интеграла для нормированной величины S = (x — m_x / δ_x), по которым, записав таблицу в память ЭВМ, можно находить по выражению (2. 2) случайные величины, распределенные по нормальному закону.

На практике для данного закона нашел применение другой метод, основанный на центральной предельной теореме вероятностей. Почтой теореме в результате суммирования определенного числа независимых случайных величин, сравнимых по первым двум моментам распределения получается случайная величина, приближенно распределенная по нормальному закону.

Принятый алгоритм получения нормально распределенной величины X заключается в следующих операциях:

1. Производится розыгрыш *N* случайных равномерно распределенных величин r_i (обычно N >= 12).

2. Находится сумма N величин $S = \sum_{i=1}^{N} r_i$, математическое ожидание которой $m = \sum_{i=1}^{N} r_i \cdot P_i = N \cdot 0{,}5$, дисперсия $D = \sum_{i=1}^{N} (r_i - m)^2 \cdot \frac{1}{N} = 1$ и среднеквадратичное отклонение $\sigma = \sqrt{D} = 1.$

3. Производится нормирование величины S_H = (S-m)/σ = S— 0,5N.

4. Находится значение случайной величины X, распределенной по нормальному закону, исходя из обратной нормированной величины

формулы

$X = \sigma_x S + m$ или

$$x_i = \sigma_x (\sum_{i=1}^{N} r_i - N/2) + m_x. \qquad (2.7)$$

Блок-схема алгоритма моделирования нормально распределенной случайной величины приведена на рис. 2.10.

2.3.6. Моделирование пуассоновского потока

Закону Пуассона подчиняются многие процессы — число сообщений, поступающих на телефонные, телеграфные станции, число заказов на вычислительный центр, число заявок на обслуживание на предприятиях быта и т.д. Закон Пуассона описывает дискретные события, выражаемые числом поступивших заказов, заявок. Он тесно связан с показательным законом, а именно, если время между поступлениями двух сложных заявок описывается доказательным законом, то число таких заявок за определенный/интервал времени описывается законом Пуассона. Для него характерны следующие свойства.

1. Стационарность, когда вероятность попадания того или иного числа событий на участок X (например, на участок времени τ) зависит только от длины участка ν и не зависит от места расположения данного локального участка на общем участке X.

2. Ординарность, при которой вероятность попадания двух или более заявок на элементарный участок $\Delta X \to 0$ пренебрежимо мала по сравнению с вероятностью попадания одного события.

3. Отсутствие последствия, когда для любых двух неперекрывающихся участков времени число событий, попадающих на один из них, не зависит от числа событий, попадающих на другие.

Поток Пуассона играет среди потоков сообщений заявок особую роль. Известно, что при суммировании большого числа случайных величин, получается величина, распределенная по нормальному закону. Аналогично, при наложении большого числа ординарных стационарных потоков с любым последствием получается пуассоновский поток.

Поток Пуассона характеризуется следующим выражением для вероятности наступления К событий:

$$P(K) = \frac{a^K}{K!} \cdot e^{-\lambda X}, \qquad (2.8)$$

где К — число событий, а — параметр потока.

Для участка х а = λх , и где λ — интенсивность наступления событий. Соответственно

$$P(K) = \frac{(\lambda x)^K}{K!} \cdot e^{-\lambda X}. \qquad (2.9)$$

Для вероятности, когда участок Х окажется пустым, т.е. К = 0, выражение (2.9) принимает вид

$$P(0) = P_0 = e^{-\lambda x}. \qquad (2.10)$$

Из (2.10) следует, что вероятность P_0 является противоположным событием по отношению к интегральной функции распределения случайной величины по показательному закону

$$P_0 = 1 - F(x).$$

Математическое ожидание и дисперсия такого распределения определяются выражениями:

$$m = \int_0^\infty x f(x) dx = \int_0^\infty \lambda x e^{-\lambda X} dx = 1/\lambda,$$

$$D = \int_0^\infty x^2 f(x) dx - m^2 = \int_0^\infty \lambda x^2 e^{-\lambda X} dx - 1/\lambda^2,$$

откуда

$$D = 1/\lambda^2 \qquad (2.12)$$

$$\sigma = 1/\lambda. \qquad (2.13)$$

Из свойства отсутствия последствия пуассоновского потока следует аналогичное свойство показательного закона. Если промежуток времени, распределенный по показательному закону, уже длился некоторое время τ, то это никак не влияет на закон распределения оставшейся части промежутка $F^{(\tau)}(t)$

$$F(t) = F^{(\tau)}(t) = 1 - e^{-\lambda t}. \qquad (2.14)$$

Имитационное моделирование (розыгрыш) числа (К) событий, образующих пуассоновский поток с параметром λ на участке Х производится в следующей последовательности (при начальном задании исходных данных λ, К=0).

Разыгрывается случайное число r_1, по которому находится величина X_1.

Согласно выражению (2.3), $X_1 = —(1/\lambda) \ln r_1$.

Осуществляется проверка $X_1 < X$.

Вычисление этого условия означает, что первое событие попало внутрь участка Х. Текущее значение счетчика числа попавших внутрь участка Х событий К увеличивается на 1 (К=1). Затем разыгрывается следующее случайное число r_2 , по которое находится X_2. Проводится проверка условия ($X_1 + X_2$) < Х . При выполнении этого условия значение К вновь увеличивается (К=2) и процесс вновь повторяется.

При нарушении условия $\sum X_i < X$ фиксируется разыгранное значение числа событий, попавших на участок Х (К = i). Блок-схема алгоритма описанного процесса приведена на рис.2.11.

2.3.7. Моделирование потока Эрланга

Поток Эрланга получается путем "просеивания" пуассоновского потока. Если взять пуассоновский поток и выбросить из него каждую вторую точку (рис. 2.12,а), то оставшиеся точки образуют поток Эрланга первого порядка. При выбрасывании подряд двух точек получается поток Эрланга второго порядка (рис. 2.12, б) и т.д. Поток Пуассона можно рассматривать как поток Эрланга нулевого порядка. Из рассмотренных трех свойств простейшего (пуассоновского) потока для потока Эрланга остаются свойства

ординарности и стационарности, но не выполняется свойство − отсутствие последствия, так как интервалы T_I, T_2 (рис. 2.12) уже не описываются показательным распределением.

Для потока Эрланга К порядка расстояние между двумя смежными событиями

$$T = \sum_{i=1}^{K+1} T_i ,$$

где T_i — интервал между событиями пуассоновского потока с плотностью распределения $f(t) = \lambda e^{-\lambda t}$, $(t > 0)$.

Для получения плотности распределения потока Эрланга (Эк) К порядка $f_k(t)$ можно исходить из того факта, что интервал Т для потока Эк с вероятностью $f_k(t)dt$ примет значение между t и dt (рис. 2.12, б). По отношению к пуассоновскому потоку это равносильно попаданию К его точек внутри участка (0, t), а последней точки — на участок (t, t + dt). Вероятности этих событий соответственно равны

$$P_k(t) = \frac{(\lambda t)^k}{k!} \cdot e^{-\lambda t}$$

и λdt.

Отсюда

$$f_k(t)dt = \frac{\lambda(\lambda t)^k}{k!} \cdot e^{-\lambda t} dt, \quad f_k(t) = \frac{\lambda(\lambda t)^k}{k!} \cdot e^{-\lambda t}. \qquad (2.15)$$

Очевидно, при К = 0

$f_0(t) = \lambda e^{-\lambda t}$,

что идентично показательному закону.

Математическое ожидание и дисперсия потока Эрланга имеют вид:

$$m_k = \sum_{i=1}^{K+1} m_0 = (K+1)m_0 .$$

С учетом

$m = 1/\lambda,$

$$m_k = \frac{K+1}{\lambda}. \qquad (2.16)$$

По теореме сложения дисперсий

$$D_k = \frac{K+1}{\lambda^2}, \qquad \sigma_k = \frac{\sqrt{K+1}}{\lambda}.$$

Плотность потока Эрланга

$$\Lambda_k = \frac{\lambda}{K+1}.$$

Таким образом, при увеличении порядка потока Эрланга увеличиваются как математическое ожидание, так и дисперсия, а плотность потока падает.

Интересно свойство потока Эрланга при К→ ∞ и сохранении его плотности (или математического ожидания). Для этого рассмотрим нормированный поток Эрланга, когда m_k = const.

Изменяя масштаб времени, получим

$$\overline{T} = \frac{T}{K+1}, \; \Lambda_k = \lambda(K+1).$$

Соответственно,

$$\bar{f}_k(t) = \frac{\lambda(K+1)}{K!}(\lambda(K+1)t)^K \cdot e^{-\lambda(K+1)t}. \tag{2.17}$$

Математическое ожидание величины $\overline{T}$, исходя из исходного условия, не зависит от К

$$\overline{m}_k = m_0 = 1/\lambda.$$

Дисперсия величины $\overline{T}$

$$\overline{D}_k = \frac{D_k}{(K+1)^2} = \frac{1}{\lambda^2(K+1)}. \tag{2.18}$$

Из (2.18) следует, что дисперсия неограниченно убывает с возрастанием К.

Отсюда следует важный для практических приложений вывод: при неограниченном увеличении К нормированный поток Эрланга приближается к регулярному потоку с постоянными интервалами.

Таким образом, задаваясь различными значениями К, можно получить поток событий с любой степенью последействия, начиная от полного отсутствия (К = 0) до жесткой связи между моментами появления

событий (К = ∞).

Наиболее естественным и простым вариантом моделирования потока Эрланга является использование его физической сути, а именно, период потока Эрланга Т является суммой К периодов пуассоновского потока. В таком алгоритме необходимо реализовать моделирование случайных величин появления потоков событий для пуассоновского потока.. Затем К таких величин складываются, в результате чего образуется случайная величина Т, характеризующая время между моментами появления смежных событий для потока Эрланга К порядка. Блок-схема алгоритма для этого варианта приведена на рис. 2.13.

2.3.8. Моделирование гиперэкспоненциального распределения

Для потока Эрланга относительная вариация

$$q = \frac{\sigma^2}{\tau^2} \le 1. \qquad (2.\ 19)$$

Обобщением экспоненциального (показательного) распределения на случай q ≥ 1 является гиперэкспоненциальное распределение, интегральная функция которого

$$F(x) = 1 - \sum_{i=0}^{K-1} a_i \cdot e^{-\alpha_i x}, \quad \sum_{i=0}^{K-1} a_i = 1. \qquad (2.\ 20)$$

Интервалы между моментами наступления смежных событий потока с гиперэкспоненциальным распределением описываются экспоненциальными распределениями, но с различными значениями параметра распределения для отдельных интервалов α_i.

Гиперэкспоненциальное распределение можно интерпретировать следующим образом. После наступления очередного события момент наступления следующего события может быть разыгран на узле, состоящем из К параллельных фаз. Вероятность попадания на i фазу равна a_i. Время наступления следующего события на i фазе распределено по экспоненциальному закону с параметром α_i. При сохранении свойства ординарности,

одновременно на всех фазах не может находиться больше одной заявки.

Исходя из приведенного объяснения физической сути гиперэкспонениального распределения, приводится алгоритм моделирования этого закона (рис. 2.14). Исходными данными для модели являются порядок распределения К, вероятность попадания на i фазу (из имеющихся К параллельных фаз) a_i , параметр экспоненциального распределения на i фазе λ_i. Значения a_i и λ_i задаются в виде одномерных массивов исходных данных с объемом К А(K), Λ(K). Выбор соответствующей фазы осуществляется путем розыгрыша вероятности попадания случайной величины r на заданный участок в интервале 0, 1, разбитый на неравномерные участки a_i (блоки 2-6, рис. 2.14). После определения i фазы обслуживания осуществляется генерация случайного числа по экспоненциальному распределению с параметром λ_i, (блоки 7, 8).

Пример. Исходные данные : К=6

Значения А(К), Л(К) приведены в табл. 2.1

Таблица 2.1

	I	2	3	4	5	6
А (К)	0,2	0,15	0,18	0,16	0,22	0,09
λ (К)	0,3	0,6	0.08	0,36	0,7	0,5

Пусть в результате розыгрыша случайной величины r, равномерно распределенной в интервале (0, 1) , получилось значение r = 0.6 (блок 3).

В начале выполнения алгоритма i = 1, A(1) = 0,2. Сравнивая r и S = A(1), получаем r>S (блок 4). Поэтому далее выполняются блоки 5, 6 (i = 2, S = S + A(2) = 0,35) и вновь производится сравнение r и S. Эта цепь алгоритма выполняется до нарушения условия r>S, когда для данного примера i = 4, S = 0,69. С этого момента начинают выполняться блоки 7, 8, а именно, разыгрывается новое значение равномерно распределенной случайной величины и определяется интервал до момента времени наступления нового события по экспоненциальному распределению с параметром выбранной фазы λ (4) = 0,36. Полученная случайная величина

затем используется в основной программе имитационного моделирования исследуемого процесса.

2.4. Моделирование цепей Маркова с дискретным временем

В предыдущем пункте были рассмотрены алгоритмы моделирования случайной величины, распределенной по тому или иному закону. Под случайной величиной понималась величина, которая в результате опыта может принять то или иное значение. Неизвестно заранее, какое именно. На практике часто приходится иметь дело со случайными величинами, изменяющимися в процессе исследований (опыта). Такие случайные величины, которые в процессе опыта изменяются и могут принять тот или иной конкретный вид, неизвестно заранее какой, называются случайными функциями. Изучению случайных функций посвящена отрасль теории вероятностей — теория случайных процессов.

Из множества видов исследованных ныне случайных процессов важное место занимают марковские процессы, которые характеризуются свойством отсутствия последействия. Это свойство проявляется в том, что для предсказания вероятностного характера в будущем достаточно знать состояние процесса в настоящий момент независимо от того, когда и как процесс перевел в это (настоящее) состояние.

Различают марковские процессы с дискретным и непрерывным временем. Цепью Маркова с дискретным временем называют цепь, изменение состояний которой происходит в определенные дискретные моменты времени, а в цепи Маркова с непрерывным временем изменения состояний могут происходить в любые случайные моменты.

Рассмотрим систему, состоящую из трех измерительных комплектов (Ж), контролирующих параметры технологического процесса. Результаты измерений подаются в центральный управляющий вычислительный комплекс АСУ через фиксированные интервалы времени Т. В момент передачи измерительных сигналов система может находиться в различных состояниях, характеризующихся количеством неисправных ИК (рис. 2. 15).

Состояние E_0 соответствует исправному состоянию всех трех ПК. За время Т система может остаться в этом состоянии с вероятностью P_{00} или же перейти в другие состояния E_i, (i =1, 2, 3), при которых неисправны i измерительных комплектов с соответствующими вероятностями P_{0i}. При нахождении системы в состоянии E_i она может перейти в другое состояние с вероятностью P_{ij} или остаться в прежнем состоянии с вероятностью P_{ij}.

Система считается работоспособной, если исправны не менее двух приборов. В этом случае из трех потоков сигналов методом выбора по большинству выбираются идентичные сигналы, которые принимаются за истинные. При наличии хотя бы двух исправных ПК измерительные сигналы с их выходов будут идентичными. Сигналы третьего ИК при этом в расчет приниматься не будут. Такая система является реализацией так называемого метода мажоритарного резервирования, применяемого для повышения надежности работы системы. Очевидно, состояния E_0, E_1 будут соответствовать случаю, когда система работоспособна, а состояния E_2, E_3 — случаю отказа системы.

Описанный процесс является марковским с дискретным временем переходов из одного состояния в другое. Следует иметь в виду относительную условность такого рассмотрения. Отказы в отдельных ИК могут наступать в любые случайные непрерывные моменты времени. Если эту систему рассматривать с точки зрения организации ремонтных работ, то она должна рассматриваться как цепь Маркова с непрерывным временем, когда сразу же после возникновения неисправности необходимо приступать к восстановлению отказавшего комплекта. В рассматриваемом же случае нас интересуют лишь состояния системы в фиксированные моменты времени, когда надо передавать результаты измерений. Промежуточные состояния и переходы системы внутри интервала $t_i - t_{i+1}$ ($t_{i+1} - t_i = T$), нас не интересуют. С этой точки зрения рассматриваемая система относится к цепи Маркова с дискретным временем. Таким образом, рассматривае

мая система, в зависимости от методов ее анализа, может рассматриваться как цепь Маркова с непрерывным или дискретным временем.

Для определения марковской цепи необходимо знать совокупность начальных состояний P_i, соответствующих нахождению системы в начальный момент времени в состоянии E_i. Кроме того, для установления зависимостей, связывающих каждую пару состояний (E_i , E_j), составляется матрица вероятностей переходов:

$$P = \begin{vmatrix} P_{11} & P_{12} & \dots & P_{1n} \\ P_{21} & P_{22} & \dots & P_{2n} \\ \dots & \dots & \dots & \dots \\ P_{n1} & P_{n2} & \dots & P_{nn} \end{vmatrix},$$

где $\sum_{j=1}^{n} P_{ij} = 1$ для каждого фиксированного значения *I,* так как каждая строка матрицы состоит из вероятностей, составляющих полную группу событий переходов из состояния i в любое возможное состояние j, включая j = n.

Данная матрица называется стохастической. Если характеристики установившегося режима не зависят от начальных вероятностей P_i (0), то такая марковская цепь называется эргодической.

Процесс моделирования рассмотренной цепи Маркова состоит в получении последовательности состояний E_1 , E_2 ... E_n в соответствии с заданной матрицей переходов. Алгоритм моделирования будет основан на определении марковской цепи, а именно на том, что исход каждого последующего перехода зависит только от результата предыдущего. При моделировании необходимо взять (n+1) участков (0, 1). Первый участок разбивается на отрезки в соответствии с вероятностями начальных состояний системы P_i (0), $\sum_{i=1}^{n} P_i(0) = 1.$ Остальные участки разбиваются на отрезки в соответствии с переходными вероятностями, соответствующими строкам матрицы P_{1i} , P_{2i} , ... , P_{ni} (рис. 2.16).

При моделировании вначале определяется начальное состояние

системы. Для этого разыгрывается случайное число r, равномерно распределенное на участке (0, 1), Затем, согласно вышеописанной процедуре моделирования дискретной случайной величины, определяется состояние S_k, исходя из условия $P_к(0) < r < P_{k+1}(0)$ для первого участка. Вновь разыгрывается следующее случайное число и сравнивается с вероятностями К, строки матрицы P_{ki}. Состояние системы не изменится при условии $P_{к,к} < r < P_{к,к+1}$ или осуществится переход в следующее состояние при условии

$P_{KJ} < r < P_{KJ+I}$.

Дальнейшие шаги моделирования идентичны. При большом числе испытаний N система L_k раз будет находиться в К состоянии. Отношение L_k/N будет соответствовать вероятности К состояния ($P(S_k) = L_k/N$). Коэффициент готовности системы для рассматриваемого примера будет определяться суммой вероятностей $K_r = P(S_0) + P(S_1)$.

Таким образом, путем статистического моделирования можно определить вероятности отдельных состояний дискретной цепи Маркова, а на основе этих вероятностей — необходимые характеристики системы.

2.5. Определение объема имитационных экспериментов

Объем моделирования определяется характером исследуемых величин (случайных событий, величин или процессов), требуемой точностью, конкретными условиями проведения моделирования.

Определим необходимое число экспериментов при оценке вероятности наступления события, оценкой которой является частота $P^* = L / N$, где L — количество успешных опытов в процессе моделирования, N — общее число испытаний. Частота P^* является случайной величиной, так как она будет принимать разные значения при повторении серии опытов N. Согласно предельной теореме теории вероятностей, случайная величина P^* распределена приблизительно по нормальному закону.

Введем дискретную случайную величину Z с законом распределения $P(Z = 1) = P^*$, $P(Z = 0) = 1 — P^*$.

Математическое ожидание и дисперсия величины Z равны

$M = 1 \cdot P^* + (1 - P^*) \cdot D = P^*$,

$D = P^*(1 - P^*)^2 + (1 - P^*) \cdot (0 - P^*)^2 - P^*(1 - P^*)$.

Затем определим математическое ожидание и дисперсию случайной величины

$$X = \frac{1}{N}\sum_{i=1}^{N} Z_i, \tag{2.22}$$

$$M(X) = M(\frac{1}{N}\sum_{i=1}^{N} Z_i) = \frac{1}{N}\sum_{i+1}^{N} M(Z_i) = P^*,$$

$$D(X) = D(\frac{1}{N}\sum_{i=1}^{N} Z_i) = \frac{1}{N^2}\sum_{i+1}^{N} D(Z_i) = \frac{1}{N} P^*(1 - P^*),$$

$$\sigma(X) = \sqrt{\frac{P^*(1 - P^*)}{N}}. \tag{2.23}$$

Найдем число испытаний, при котором значение X отличается от вероятности P^* меньше, чем на заданную величину ε с заданной достоверностью α.

Для нормированной величины X

$$\overline{X} = \frac{X - m_X}{\sigma_X} = \frac{X - P^*}{\sqrt{P^*(1 - P^*)/N}}. \tag{2.24}$$

С учетом нормального закона распределения величины $\overline{X}$ выражение (2. 23) будет иметь вид

$$P(-U_\beta < \frac{X - P^*}{\sqrt{P^*(1 - P^*)/N}} < U_\beta) = \alpha, \tag{2.25}$$

где $\beta = (1-\alpha)/2$ — уровень значимости;

U_β — квантиль, соответствующий значению $\beta = P(\overline{X} > U_\beta)$.

Значения β и U_β табулированы. Например, при $\beta = 0{,}61$, $U_\beta = 1{,}28$, $\beta = 0{,}025$, $U_\beta = 1{,}96$ и т.д.

Из (2.25) получим $N = (U_\beta^2 \cdot P^*(1 - P^*))/\varepsilon^2$. (2.26)

При моделировании величина P^* обычно неизвестна. Поэтому вначале проводится моделирование объемом N = 50 – 100 выборок, по

которому определяется P^*, а затем из (2. 26) окончательно находится N.

В табл. 2.2. приводится необходимое число реализации для получения оценки L / N с точностью ε и достоверностью $\alpha = 0{,}95$ для различных значений P^*.

Таблица 2.2

P^*		ε		
		0,05	0,02	0,01
0.2	0.8	250	1500	6200
0,3	0,7	330	2100	8400
0,4	0.6	380	2300	9400
0,5		390	2400	9800

Приведенные в табл. 2.2 значения требуемого объема имитационных экспериментов даже при самих простых оценках показывают необходимость проведения большого числа реализации на ЭВМ.

Для каждых конкретных случаев моделирования целесообразно проводить такую оценку и определять погрешность вычислений в зависимости от реализованного объема моделирования.

3. РЕШЕНИЕ ЗАДАЧ, ОПИСЫВАЕМЫХ НЕПРЕРЫВНО-СТОХАСТИЧЕСКИМИ МОДЕЛЯМИ

3. 1. Основные принципы моделирования непрерывно-стохастических систем

Непрерывно-стохастические системы наиболее характерны для исследования вероятностных моделей компьютерных сетей.

В настоящее время разработан ряд принципов построения алгоритмов для моделирования таких систем. В общем, все алгоритмы данного типа можно разделить на две группы: детерминированные и стохастические.

В детерминированных алгоритмах используется так называемый "принцип Δt", по которому состояние системы последовательно рассматривается через определенные (детерминированные) интервалы времени Δt. Каждое новое состояние блоков системы определяется в момент времени $t+\Delta t$ в соответствии с действующими случайными факторами, задаваемыми распределениями их вероятностей.

Данный принцип универсален, может быть реализован набором стандартных программных блоков. После определенного практического навыка на первоначальном периоде освоения этого принципа, процесс составления алгоритмов и программ при реализации различных моделей по Δt — принципу может быть значительно упрощен. Однако этот принцип неэкономичен, поскольку на каждом шаге необходимо просматривать состояние всех узлов и блоков системы, несмотря на то, что большинство (а то и все) этих блоков за время Δt не изменяют своего состояния. Если же увеличивать интервал Δt, то появляется опасность пропуска отдельных событий в системе.

Сам характер процессов, происходящих в непрерывно-стохастических системах, исходит из того факта, что состояние отдельных ее узлов изменяется с моментом времени, совпадающего с наступлением определенных событий — поступлением очередной заявки на обслуживание, окончания ее обработки, момента выхода из строя

оборудования и т.п. Поэтому моделирование таких систем удобно проводить, используя событийный принцип. Этот принцип заключается в том, что текущее время в моделируемой системе изменяется дискретно, проходя последовательно через все события, т. е. моменты изменения состояния блоков системы. Момент наступления следующего события в системе определяется минимальным значением моментов времени из их списка будущих событий, которые будут происходить в каждом блоке системы.

В процессе имитационного моделирования этих потоков заявок можно определить случайные времена наступления первого и второго события при каждой реализации (соответственно t_1 и t_2). При $t_1<t_2$ принимается событие, когда в систему поступает новая заявка. В модели после этого реализуются шаги, связанные с последствием наступления такого события. Одновременно разыгрывается новое время t'_1 поступления следующей заявки. На следующем шаге будут сравниваться величины t'_1 и t_2 . Таким образом, на каждом шаге моделирования выбирается событие с минимальным временем его наступления по отношению к другим событиям, которые могут иметь место в системе.

При событийном принципе интервалы времени, в которые просматривается состояние системы, изменяются с переменным шагом. Кроме того, просматриваются состояния не всех узлов системы, а только тех, в которых события произошли.

Известны разновидности возможных вариантов реализации событийного принципа (или принципа ΔZ): с синхронизацией работы всех узлов системы по одному выбранному в качестве ведущего «или без синхронизации с рассмотрением состояния отдельных узлов автономно, независимо от других, с циклическим или спорадическим (выборочным) просмотром состояний отдельных узлов. Целесообразность выбора того или иного варианта алгоритма обуславливается видом моделируемой системы.

Ниже приводятся отдельные имитационные модели непрерывно-

стохастических систем в основном на основе событийного принципа, имеющего вышеприведенные преимущества по сравнению с принципом Δt.

3.2. Моделирование процессов в одноканальной системе массового обслуживания с отказами

В таких системах заявка, поступившая в момент времени, когда обслуживающий прибор занят, получает отказ и покидает систему.

Пусть на обслуживающий прибор поступает поток заявок с заданным законом распределения длительности интервалов поступления этих заявок. Каждая заявка обслуживается на рабочем месте за время τ также в соответствии с определенным законом распределения. Если поступающая заявка застает прибор (робот) занятым, то она получает отказ в обслуживании (рис. 3.1). Результатом исследования в данном случае является оценка числа обслуженных заявок и заявок, получивших отказ в обслуживании.

Блок-схема алгоритма данной модели (рис. 3.2) составлена с использованием событийного принципа, изменение состояний системы вызывается наступлением двух видов событий — моментом поступления новой заявки t_1 или моментом окончания обслуживания очередной заявки t_2. Блок минимизации и выбора минимального времени в этой системе сводится к сравнению этих двух значений времени.

В блок-схеме алгоритма используются следующие обозначения:

t_1 — момент времени поступления новой заявки на обслуживание;

t_2— момент времени окончания обслуживания очередной заявки;

$T_м$ — общее время моделирования;

Z — состояние системы (при $Z=0$ система свободна, при Z=1 система обслуживает очередную заявку);

$\tau_з$ — интервал времени между моментами поступления смежных заявок согласно заданному закону ;

$\tau_{об}$— время обслуживания заявки;

N_n , $N_{об}$, N — соответственно количество потерянных, обслуженных и общее число заявок.

В начале моделирования задаются исходные данные $t_1 = \infty$, $t_2 = \infty$ (заявки не поступают и отсутствуют на обслуживании), $T_м$ =0,=Z=0, N=0 (блок 2). Затем, в соответствии с принятым законом поступления потока заявок, моделируется момент времени поступления первой заявки $t_1 = \tau_з$ (блок 3). Например, для пуассоновского потока заявок с параметром λ, согласно (2.3), моделируемое значение $\tau_з = — (1/ \lambda) \ln r_i$. При первом сравнении $\tau_1 = \tau_3 < \tau_2$ (блок 4). Поэтому управление передается на блок 5, в котором значение количества поступивших заявок N увеличивается на 1. В блоке 6 осуществляется проверка состояния системы. На первом шаге (испытании) система свободна (Z = 0). В этом случае начинается обслуживание поступившей заявки, заключающееся в формировании времени обслуживания $\tau_{об}$ и текущего момента времени окончания обслуживания $t_2=t_1+\tau_{об}$. При поступлении заявки на обслуживание система переходит в занятое состояние (Z = 1, блоки 7-9). Затем в блоках 10,11 моделируются интервал времени между поступлениями обслуживаемой и следующей заявкой τ_3 и текущий момент времени поступления следующей заявки $t_1=t_1+\tau_3$.

На следующем шаге сравниваются сформированные значения t_1, и t_2. Если следующая заявка поступит раньше окончания текущей заявки ($t_1 <t_2$, Z = 1), то она получает отказ в обслуживании. Значение потерянных заявок N_n, увеличивается на 1 (блок 12). Затем в блоках 13, 14 моделируются новые значения τ_3, t_1. Новое значение t_1 будет теперь сравниваться с прежним значением t_2 . Рассмотрим случай $t_2 < t_1$. Каждый раз по окончании обслуживания очередной заявки необходимо сравнивать значение *i,* с общим временем моделирования (блок 15). При $t_2 < T_м$ число обслуженных заявок $N_{об}$ увеличивается на 1 (блок 17) и устанавливаются исходные данные, соответствующие свободному состоянию системы (Z = 0, $t_2 = \infty$, блоки 16,19). При достижении заданного времени моделирования ($t_2 > T_{мод}$) на печать и обработку выводятся необходимые

параметры N , N_n , $N_{об}$, вероятность потерь $P_n = N_n / N$. Учет времени пребывания $\tau_{преб}$ заявки в рассмотренной системе не нужен ввиду отсутствия режима ожидания, т.е. $\tau_{преб} = \tau_{об}$.

3.3. Моделирование процессов в многоканальной системе с отказами

В многоканальной системе одновременно могут обслуживаться n заявок, где n — число обслуживающих приборов, устройств, например, группа однотипных станков, линий и другого оборудования, выполняющих идентичные операции. Любая следующая заявка, заставшая все приборы занятыми, получает отказ в обслуживании. На примере трехприборной системы (рис. 3.3) показан случай, когда во время обслуживания первой заявки, поступившей в момент времени t_1^1, в систему поступило две заявки в моменты t_1^2, t_1^3, которые начали обслуживаться соответственно вторым и третьим приборами (устройствами, станками). В момент времени t'_2, первый прибор освободился, но во время занятого состояния второго и третьего приборов на него поступила следующая заявка. При наличии в системе не более трех заявок отказов от обслуживания не происходит. Однако при поступлении в систему любой заявки, заставшей все три прибора занятыми (моменты t_1^4, t_1^5), они теряются.

На рис. 3.4 показана блок-схема алгоритма моделирования n — канальной системы с потерями. Принятые обозначения:

n — количество обслуживающих приборов;

$t_1(n)$ — массив времени поступления заявок на отдельные приборы;

$t_2(n)$— массив времени окончания обслуживания заявок на отдельных приборах;

Z (n)— массив состояний отдельных приборов (при $Z(i) = 0$ i прибор свободен, а при $Z(i) = I$ — занят).

Остальные обозначения (t_1 , t_2 , N_n , $N_{об}$, N, $T_м$, $\tau_з$, $\tau_{об}$) соответствуют обозначениям одноприборной модели.

В начальный момент времени все приборы свободны (Z (i) = 0,

$i=1,n)$, $t_1(n)=t_2(n)=\infty$, $N_{об}=N_n=0$ (блок I).

Первая сгенерированная заявка поступает в первый прибор ($t_1 = \tau_з$, блоки 2, 3).

В рассматриваемой модели любая сгенерированная заявка или сразу же начинает обслуживаться в i приборе, или теряется. Соответственно в блоке выбора минимальных времен, соответствующих смене ситуаций в системе, фигурирует только одно значение времени t_1, соответствующее текущему моменту времени появления следующей заявки. В то же время в разных приборах могут одновременно обслуживаться несколько заявок. Таким образом, в блоке выбора минимального времени должны сравниваться значения времени t_1 и $t_2(i)$. Выбор меньшего из них может производиться путем их сортировки, т.е. записи в порядке возрастания их значений (блок 4).

В случае $t_1=\min$, в блоках 6-9 производится определение прибора с меньшим номером, свободного от обслуживания ($Z(i) = 0$). Если $i \le N$, то заявка принимается на обслуживание в i прибор ($Z(i) = I$, блок II). Процесс обслуживания заключается в генерации времени обслуживания $\tau_{об}$, определения момента конца обслуживания $t_2(i)=t_1(i)+\tau_{об}$ генерации времени до поступления следующей заявки τ_3 и текущего момента времени поступления этой заявки $t_1(i)=t_1(i)+\tau_з$ (блоки 12-15). Сформированные значения t_1 и $t_2(i)$ поступают в блок 4 выбора минимальных времен, где производится сортировка массива с учетом поступивших новых значений времени. В случае перебора состояний приборов окажется, что все они заняты ($i > N$ блок 10), то поступившая заявка теряется ($N_n= N_n+1$ блок 16). Затем генерируется время поступления следующей заявки t_1, (блоки 17, 16).

При минимальном времени окончания обслуживания заявки в i приборе ($t_2(i) = \min$, $t_2(i) < T_м$) фиксируется число обслуженных заявок ($N_{об} = N_{об} + 1$), находится время пребывания заявки в системе $t_{пр} = t_2(i)—t_1(i)$, после чего устанавливаются исходные данные для освободившегося прибора ($Z(i) = 0$, $t_1(i) = t_2(i) = \infty$) (блоки 21-24). Следует заметить, что

блок 22, в котором определяется время пребывания заявки в системе, может быть опущен, так как это время будет соответствовать задаваемому времени $\tau_{об}$ **.** Введение этого блока позволяет контролировать соответствие вырабатываемых ЭВМ случайных чисел с заданным законом.

По окончании моделирования ($t_2 > T_м$) осуществляется вывод на печать полученных результатов — N_n , $N_{об}$, N, $P_n = N / N_n$, гистограммы распределения $t_{пр}$ в случае надобности.

3.4. Моделирование процессов в одноканальной системе с ограниченным ожиданием

В системе с ожиданием поступающие заявки имеют возможность ожидать начала своего обслуживания. Например, поступающие для обработки на станке заготовки и т.п. Обычно на процесс ожидания накладываются определенные ограничения, связанные с недопустимостью превышения длины очереди заявок или времени ожидания в очереди (длиной максимального интервала конкретной операции).

На рис. 3.5 показан временной процесс обслуживания заявок в одноканальной системе, когда имеется один обслуживающий прибор. Ограничения на ожидание определяются допустимой длиной очереди M_g=2. Поступающая в момент времени t_1 первая заявка застает обслуживающий прибор ОП свободным и сразу же начинает обслуживаться, время начала ее обслуживания t_1'' равно времени ее поступления в систему $t_1'' = t_1$. Поступающие заявки образуют поток событий с заданным законом распределения. Время занятости обслуживающего прибора τ (длительность обслуживания) является случайной величиной, также описываемой конкретным законом распределения. Заявки в системе обслуживаются в порядке очереди, т.е. в том порядке, в котором они поступили в систему. Для рассматриваемого случая первая заявка успевает отслужиться в момент времени $t_1^{об}$ раньше поступления следующей заявки. Поэтому поступающая в момент t_2 вторая заявка застает обслуживающий прибор свободным и сразу же поступает на обслуживание $t_2'' = t_2$. Во время обслуживания этой заявки

дли

тельностью τ_2 в моменты времени t_3 , t_4 , t_5 поступают следующие заявки, которые застают ОП занятым. Третья и четвертая заявки дожидаются освобождения прибора от обслуживания предыдущей заявки, после чего сразу же принимаются на обслуживание $t_3'' = t_2^{об}$, $t_4'' = t_3^{об}$. Для пятой заявки нарушается условие допустимой длины очереди $M_g = 2$, так как эта заявка поступила еще в момент обслуживания второй заявки. В результате этого эта заявка получает отказ в обслуживании. Следующая шестая заявка поступает в момент обслуживания третьей заявки, т.е. становится второй в очереди за четвертой заявкой. Следовательно, она может ожидать начала своего обслуживания.

Один из вариантов блок-схемы алгоритма для реализации системы с ограниченным ожиданием на ЭВМ приведен на рис. 3.6. В первом блоке этой модели задаются необходимые исходные данные — число поступивших заявок i = 0, число обслуженных и потерянных заявок $N_{об} = 0$, $N_n = 0$, допустимая длина очереди M_g , начальные значения времени поступления и обслуживания $t_0 = 0$, $t_0^{об} = 0$. Необходимо задать также общее время

моделирования на ЭВМ $T_м$, вид и параметры входящего потока заявок и времени обслуживания. Например, для пуассоновского потока заявок при экспоненциальном распределении времени обслуживания одной заявки задается интенсивность пуассоновского потока λ и интенсивность обслуживания μ. В блоке 2 предусматривается реализация подпрограммы формирования времени генерации очередной заявки τ_3, в соответствии с заданным потоком распределения. Примеры таких подпрограмм были рассмотрены в главе 2. Блоки 3, 4 отражают количество i и момент времени поступления в систему очередной заявки $t_i = t_{i-1} + \tau_3$. После формирования очередной заявки осуществляется проверка на достижение в ЭВМ заданного времени моделирования ($t_1 > T_м$?, блок 5). При невыполнении этого условия осуществляется следующая проверка сравнения времени поступления очередной заявки со временем окончания обслуживания предыдущей заявки ($t_i < t^{об}_{i-1}$, блок 6). При выполнении этого условия, т.е.

при занятом состоянии ОП, проверяется условие превышения допустимой длины очереди ($i > M_g$, блок 7). Если очередь выше допустимой, то вновь поступившая заявка теряется, что фиксируется в счетчике потерянных заявок ($N_n = N_n+1$, блок 8). Для продолжения работы модели условно принимается, что момент времени окончания обслуживания потерянной заявки $t_1^{об}$ равен аналогичному времени предыдущей заявки $t_i^{об} = t_{i-1}^{об}$. Затем осуществляется возврат к формированию следующей заявки. При наличии очереди заявок в допустимых пределах ($i \leq M_g$). время начала обслуживания поступившей заявки принимается равным времени окончания обслуживания предыдущей заявки ($t_i'' = t_{i-1}^{об}$, блок 10). Определение t_i'' осуществляется также в блоке II, переход к которому происходит в ситуации, когда вновь поступившая заявка застает ОП свободным (при условии $t_i \geq t_{i-1}^{об}$). В этом случае заявка сразу же принимается на обслуживание ($t_i'' = t_i$, блок II). В блоках 12-15 реализуется процесс обслуживания очередной заявки. В блоке 12, аналогично блоку 2, реализуется подпрограмма формирования времени обслуживания $\tau_{об}$ в соответствии с заданным временем его распределения, в блоке 13 определяется момент окончания времени обслуживания $t_i^{об} = t_i'' + \tau_{об}$, в блоке 14 находится общее время пребывания заявки в системе, включая время ее ожидания и время обслуживания $\tau_i = t_i^{об} — t_i$. Блок 15 осуществляет подсчет общего числа обслуженных заявок $N_{об} = N_{об} + 1$. После этого происходит переход к генерации следующей заявки. По окончании общего времени моделирования ($t > T_м$) осуществляется обработка полученных результатов, которая заключается в реализации на *ЭВМ* типовой подпрограммы построения гистограммы распределения времени τ_i, пребывания заявки в системе, фиксации и выводе на печать числа обслуженных и потерянных заявок.

Приведенный вариант блок-схемы алгоритма системы обслуживания с ожиданием сравнительно прост, но весьма неэкономичен в отношении использования памяти ЭВМ, так как необходимо предусмотреть массивы $t(i)$, $t''(i)$, $t^{об}(i)$, $\tau(i)$ для больших значений i.

Например, при интен

сивности пуассоновского потока заявок λ=10 заявок/мин при $T_м$=I час среднее значение поступивших заявок составит i = 600. Для получения достоверных оценок моделирования, особенно при построении гистограммы τ_i, потребуется значение *i* порядка нескольких тысяч. Поэтому желательно применять алгоритмы, не требующие больших объемов хранящихся массивов, что отражено в другом варианте (рис. 3.7).

Во втором варианте используются два особых состояния системы — время поступления очередной заявки t_3 без учета ее номера и время обслуживания $t_{об}$. Выполнение условия $t_3 < t_{об}$ соответствует случаю поступления в данный момент времени очередной заявки. Вслед за этим определяется состояние обслуживающего прибора. При Z = 0 (блок 6) ОП свободен. Тогда время начала обслуживания этой заявки совпадает со временем ее поступления, а ОП переходит в занятое состояние (Z = 1, блок 7). Данное значение времени фиксируется в качестве первого элемента массива заявок T_3 (1) = t_3, блок 8. Затем выполняется подпрограмма формирования времени $\tau_{об}$ и определяется момент времени окончания обслуживания данной заявки $t_{об} = t_3 + \tau_{об}$ (блоки 9, 10), после чего осуществляется переход на формирование новой заявки.

Если поступающая заявка застает обслуживающий прибор занятым (Z ≠ 0, блок 6), то значение очереди заявок увеличивается на единицу (Z = Z+1, блок II), и осуществляется проверка на превышение допустимой очереди ($Z > M_g$? ,блок 12). При $Z > M_g$ фиксируется отказ от обслуживания поступившей заявки $N_n = N_n + 1$, после чего общая очередь Z уменьшается на 1. При $Z \leq M_g$, в массив заявок записывается значение времени поступившей заявки ($T_3(z) = t_3$, блок 13).

Выполнение условия $t_3 > t_{об}$ (блок 5) соответствует случаю наступления другого события в системе— окончанию обслуживания очередной заявки, которая в массиве заявок является первой. После этого определяется время пребывания заявки в системе $\tau_{пр} = t_{об} — t_3(1)$, сразу же определяется соответствующий интервал гистограммы, в который попадает значение $\tau_{об}$ (блок 17), фиксируется число обслуженных заявок

$N_{об} = N_{об}+1$ (блок 18), число заявок в системе Z уменьшается на I (блок 18). Затем проверяется условие освобождения обслуживающего прибора ($Z = 0$, блок 20). Ситуация Z=0 соответствует случаю, когда в момент окончания обслуживания очередной заявки другие заявки в ОП не поступили. Обслуживающий прибор становится свободным и находится с этого момента в режиме простоя, ожидая поступления следующей заявки. Поскольку время окончания обслуживания еще не поступившей заявки неизвестно, то оно условно принимается равным бесконечности ($t_{об} = \infty$, блок 21). При составлении программы на ЭВМ это время следует задавать большим числом, например, больше $T_м$. Аналогично это большое значение для $t_{об}$ задается в начале выполнения программы (блок I).

Ситуация $Z \neq 0$ (блок 20) соответствует случаю, когда по окончании обслуживания очередной заявки имеется массив других заявок на обслуживание. Сразу же по освобождении ОП он приступает к обслуживанию следующей заявки, находящейся первой в очереди. При этом длина очереди уменьшается на единицу. Соответственно, уменьшается на единицу номер каждой заявки очереди. Коррекция очереди осуществляется в блоках 22,23. Затем определяется время и момент окончания обслуживания для заявки под номером 1, и производится возврат к блоку 5 на сравнение нового времени окончания обслуживания первой заявки из очереди $t_{об}$ со временем поступления в систему следующей заявки t_3 .

По окончании заданного времени моделирования ($t_3 > T_м$,блок 4) осуществляется выдача на печать гистограммы времени пребывания заявок в системе (гист $\tau_{пр}$, блок 26), числа потерянных и обслуженных заявок (N_n, $N_{об}$, блок 27).

Второй вариант алгоритма требует значительно меньшей памяти ЭВМ, так как в нем хранится лишь один массив очереди заявок t(z) и нет необходимости запоминать массивы всех поступивших и обслуживаемых заявок, по сравнению с первым вариантом.

3.5 Определение показателей надежности сетей

Сети связи относятся к сложным системам , имеющим, как правило, внутреннюю избыточность, при которой выходы из строя отдельных узлов могут не приводить к прекращению обмена сообщениями между другими узлами сети. Ниже рассмотрен вариант оценки показателей надежности между двумя фиксированными узлами сети. Отказом считается такое сочетание вышедших узлов в сети, при котором все соединительные тракты передачи между рассматриваемыми узлами прерываются.

Известны аналитические методы оценок показателей надежности сетевых структур – полный перебор всех состояний, нахождение возможных путей, определение возможных сечений, метод особой точки, логико-вероятностные методы. Получили развитие методы решений на основе теории графов. Однако для сетей большой размерности наиболее приемлемым методом будет имитационное моделирование.

На рис.3.8 приведена блок-схема алгоритма имитационного моделирования сети без восстановления неисправностей. В качестве исходных данных вводятся интенсивности отказов отдельных узлов и соединительных путей между ними (λ_i, λ_{ij}), общее число отказов, разыгрываемых на ЭВМ в процессе моделирования $N_{мод}$, число узлов и матрица соединений между узлами.

В начале вычислений определяются возможные пути соединений между рассматриваемыми узлами.

Пример установления соединений для мостиковой схемы на основе последовательной фиксации соединений от исходного узла с непосредственно присоединительными к нему следующими узлами (построение дерева соединений) приведен на рис. 3.9, 3.10. Блок-схема алгоритма определения путей соединений приведена на рис.3.11.

Процесс моделирования продолжается до достижения заданного числа отказов $N_{мод}$. В случае отказа узла, не приводящего к отказу системы, данный узел далее не рассматривается ($\lambda = \lambda - \lambda_i$, $\lambda_i = 0$) и производится корректировка возможных путей соединений путем исключения путей, проходящих через отказавший узел или соединение. В

заключение по статистическим формулам находятся среднее время безотказной работы и вероятность безотказной работы системы T_c*, P_c*, а также гистограммы распределения массива $T_{ст}$* (i=1— $N_{мод}$).

Оценку показателей надежности для сети с восстановлением необходимо производить с учетом условий восстановления отказавшего узла (рис. 3.12-3.15).

Для рассматриваемой сети отказ наступает при одновременном выходе из строя узлов 1,3 или 2,4.

На рис. 3.12 приводится вариант неограниченного восстановления, когда одновременно могут восстанавливаться все отказавшие узлы.

Восстановление начинается сразу же после отказа любого узла, время восстановления не зависит от состояния сети. Приняты обозначения — t_{oi}, t_{bi}, текущее время наступления отказа, восстановление Z — индекс работоспособного (Z= 0) или отказового состояния (Z=1) $i^{го}$ узла , τ_i— время между окончанием восстановления и моментом следующего отказа i узла. Время отказового состояния τ системы определяется, пока ремонтируются все узлы (1 и 3, 2 и 4), вызвавшие отказовую ситуацию. При восстановлении одного из них сеть переходит в работоспособное состояние. Коэффициент готовности определяется по выражению

$$K_{\tilde{a}} = \frac{\sum_{i=1}^{N_{\grave{i}\hat{i}\ddot{a}}} \tau_{\eth\grave{a}\acute{a}_i}}{\sum_{i=1}^{N_{\grave{i}\hat{i}\ddot{a}}} (\tau_{\eth\grave{a}\acute{a}_i} + \tau_{\hat{i}\grave{o}\hat{e}_i})},$$

где $N_{мод}$— задаваемое число отказов, определяющее объем моделирования.

Условия ремонта сети, согласно рис.3.13, отличаются тем, что время отказового состояния сети определяется временем восстановления всех

отказавшихся узлов. Считается, что во время ремонта остальные узлы не отказывают.

На рис.3.14 приведен вариант восстановления системы во время плановых профилактических работ, проводимых независимо от состояния системы. Время отказового состояния системы включает в себя время ожидания ближайшей профилактики и время самой профилактики. После проведения профилактики системы полностью восстанавливается. На рис. 3.15 приведен вариант ограниченного восстановления, когда отказавшийся узел ожидает окончания ремонта предыдущего узла и время перерыва t на доставку ремонтного органа от одного узла к другому. В общем случае условия проведения ремонтных работ могут быть разнообразными для каждой конкретной системы.

На рис.3.16 приведена блок— схема алгоритма оценки показателей надежности сети с восстановлением применительно к рис.3.12.

СПИСОК ЛИТЕРАТУРЫ

1. Бусленко Н. П. Моделирование сложных систем. - М.: Наука, 1978.
2. Советов Б. Я., Яковлев С. А. М.: Высшая школа, 1985.
3. Лукьянов В. С. Решение задач в машиностроении методами имитационного моделирования: Учеб. пособие, ВолгПИ. - Волгоград, 1989.

Виктор Сергеевич Лукьянов

Георгий Валентинович Слесарев

Проектирование компьютерных сетей методами имитационного моделирования

Учебное пособие

Редактор Е. И. Кагальницкая

Темплан 2001г., поз. № 56

Лицензия ЛР № 020251 от 16.04.1996г.

Подписано в печать .Формат 60x84 1/16.

Бумага газетная. Печать офсетная.

Усл. печ. л. 4,18. Уч. – изд. л. 4,21.

Тираж 300 экз. Заказ .

Волгоградский государственный технический университет

400131 Волгоград, пр. Ленина, 28.

РПК “Политехник” Волгоградского государственного технического университета

400131 Волгоград, ул. Советская, 35.