ВВЕДЕНИЕ

При создании и проектировании сложных систем возникают многочисленные задачи, требующие знаний количественных и качественных закономерностей, свойственных рассматриваемым системам. Особенное значение приобрели так называемые *общесистемные* вопросы, относящиеся к общей структуре системы, организации взаимодействия между ее элементами, совокупному взаимодействию элементов с внешней средой, централизованному управлению функционированием элементов и т.д. Эти вопросы составляют существо так называемого системного подхода к изучению свойств реальных объектов и содержание направления, получившего название *системотехника*.

Наиболее полное и всестороннее исследование сложной системы на всех этапах разработки, начиная с этапа постановки задачи, подготовки технического задания и заканчивая внедрением системы в эксплуатацию, невозможно без методов моделирования. Именно моделирование является средством, позволяющим без капитальных затрат решить проблемы построения больших систем.

***Модель* — это объект-заместитель объекта-оригинала, обеспечивающий изучение некоторых свойств оригинала.**

Замещение одного объекта другим с целью получения информации о важнейших свойствах объекта-оригинала с помощью объекта-модели называется *моделированием.*

Моделирование может быть определено как представление объекта моделью для получения информации об этом объекте путем проведения экспериментов с его моделью. Теория замещения одних объектов (оригиналов) другими объектами (моделями) и исследования свойств объектов на их моделях называется ***теорией моделирования.*** Если результаты моделирования подтверждаются и могут служить основой для прогнозирования процессов, протекающих в исследуемых объектах, то говорят, что модель адекватна объекту. При этом ***адекватность*** модели зависит от цели моделирования и принятых критериев.

ОСНОВНЫЕ ПОНЯТИЯ ТЕОРИИ МОДЕЛИРОВАНИЯ

Понятие сложной системы.

Сложная система, составной объект, части которого можно рассматривать как системы, закономерно объединённые в единое целое в соответствии с определенными принципами или связанные между собой заданными отношениями.

Понятием Сложная система пользуются в системотехнике, системном анализе, операций исследовании и при системном подходе в различных областях науки, техники и народный хозяйства. Сложную систему можно расчленить (не обязательно единственным образом) на конечное число частей, называемое подсистемами; каждую такую подсистему (высшего уровня) можно в свою очередь расчленить на конечное число более мелких подсистем и т. д., вплоть до получения подсистем первого уровня, т. н. *элементов*.

Подсистема, т. о., с одной стороны, сама является Сложной системой из нескольких элементов (подсистем низшего уровня), а с другой стороны - элементом системы старшего уровня.

В каждый момент времени элемент Сложная система находится в одном из возможных состояний; из одного состояния в другое он переходит под действием внешних и внутренних факторов.

Задачи исследования сложных систем.

Одной из важных проблем в области разработки и создания современных сложных технических систем является исследование динамики их функционирования на различных этапах проектирования, испытания и эксплуатации. При исследовании сложных систем возникают задачи исследования как отдельных видов оборудования и аппаратуры, входящих в систему, так и системы в целом.

При проектировании сложных систем ставится задача разработки систем, удовлетворяющих заданным техническим характеристикам. Поставленная задача может быть решена одним из следующих методов:

- методом синтеза оптимальной структуры системы с заданными характеристиками;
- методом анализа различных вариантов структуры системы для обеспечения требуемых технических характеристик.

Оптимальный синтез систем в большинстве случаев практически невозможен в силу сложности поставленной задачи и несовершенства современных методов синтеза сложных систем. Методы анализа сложных систем, включающие в себя элементы синтеза, в настоящее время достаточно развиты и получили широкое распространение.

Любая синтезированная или определенная каким-либо другим образом структура сложной системы для оценки ее показателей должна быть подвергнута испытаниям. Проведение испытаний системы является задачей анализа ее характеристик. Таким образом, конечным этапом проектирования сложной системы, осуществленного как методом синтеза структуры, так и методом анализа вариантов структур, является анализ показателей эффективности проектируемой системы.

Среди известных методов анализа показателей эффективности систем и исследования динамики их функционирования следует отметить:

- аналитический метод;
- метод натуральных испытаний;
- метод полунатурального моделирования;
- моделирование процесса функционирования системы на ЭВМ.

Строгое аналитическое исследование процесса функционирования сложных систем практически невозможно. Определение аналитической модели сложной системы затрудняется множеством условий, определяемых особенностями работы системы, взаимодействием ее составляющих частей, влиянием внешней среды и т.п.

Натуральные испытания сложных систем связаны с большими затратами времени и средств. Проведение испытаний предполагает наличие готового образца системы или ее физической модели, что исключает или затрудняет использование этого метода на этапе проектирования системы.

Широкое применение для исследования характеристик сложных систем находит метод полунатурального моделирования. При этом используется часть реальных устройств системы. Включенная в такую полунатуральную модель ЭВМ имитирует работы остальных устройств системы, отображенных математическими моделями. Однако в большинстве случаев этот метод также связан со значительными затратами и трудностями, в частности, аппаратной стыковкой натуральных частей с ЭВМ.

Исследование функционирования сложных систем с помощью моделирования их работы на ЭВМ помогает сократить время и средства на разработку.

Затраты рабочего времени и материальных средств на реализацию модели оказываются незначительными по сравнению с затратами, связанными с натурным экспериментом. Результаты моделирования по своей ценности для практического решения задач часто близки к результатам натурного эксперимента.

Основной метод исследования сложных систем -- математическое моделирование, в том числе имитация процессов функционирования Сложная система на ЭВМ (машинный эксперимент).

Концепция применения методов математического моделирования для решения задачи исследования и проектирования сложных систем базируется на следующих основных принципах:

1. Для любой технической системы можно создать математическую модель, которая будет описывать необходимые свойства системы, или ряд моделей.
2. Техническую систему можно исследовать с помощью натурного эксперимента или с помощью математического моделирования.
3. Не всякий натурный эксперимент можно произвести, но всякий эксперимент можно промоделировать.
4. Инженерные решения можно принимать на основе адекватных математических моделей.
5. Для получения адекватных математических моделей необходим эксперимент.
6. Чтобы научиться разрабатывать адекватные математические модели можно применять сравнение численных результатов с теоретическими результатами на основе аналитических решений.

7. Математическая модель состоит из: уравнений, параметров, граничных условий.
8. Ошибка в любом компоненте математической модели даст ошибку в результате математического моделирования.
9. Конечным подтверждением принятого технического решения является натурный эксперимент.

Основные принципы моделирования

Принцип информационной достаточности. При полном отсутствии информации об исследуемой системе построение ее модели невозможно. При наличии полной информации о системе ее моделирование лишено смысла. Существует некоторый критический уровень априорных сведений о системе (уровень информационной достаточности), при достижении которого может быть построена ее адекватная модель.

Принцип осуществимости. Создаваемая модель должна обеспечить достижение поставленной цели исследования с вероятностью, существенно отличающейся от нуля, и за конечное время. Обычно задают некоторое пороговое значение P_0 вероятности достижения цели моделирования $P(t)$, а также приемлемую границу t_0 времени достижения этой цеди. Модель считают осуществимой, если одновременно выполнены два неравенства:

$$P(t) \geq P_0\,;\ t \leq t_0$$

Принцип множественности моделей. Данный принцип, несмотря на его порядковый номер, является ключевым. Речь идет о том, что создаваемая модель должна отражать в первую очередь те свойства реальной системы (или явления), которые влияют на выбранные показатель эффективности. Соответственно при использовании любой конкретной модели познаются лишь некоторые стороны реальности. Для более полного ее исследования необходим ряд моделей, позволяющих с разных сторон и с разной степенью детальности отражать рассматриваемый процесс.

Принцип агрегирования. В большинстве случаев сложную систему можно представить состоящей из агрегатов (подсистем), для адекватного математического описания которых оказываются пригодными некоторые стандартные математические схемы. Принцип агрегирования позволяет, кроме того, достаточно гибко перестраивать модель в зависимости от задач исследования.

Принцип параметризации. В ряде случаев моделируемая система имеет в своем составе некоторые относительно изолированные подсистемы, характеризующиеся определенным параметром, в том числе векторным. Такие подсистемы можно заменять в модели соответствующими числовыми величинами, а не описывать процесс их функционирования. При необходимости зависимость значений этих величин от ситуации может задаваться в виде таблицы, графика или аналитического выражения (формулы). Принцип параметризации позволяет сократить объем и продолжительность моделирования. Однако надо иметь в виду, что параметризация снижает адекватность модели.

Степень реализации перечисленных принципов в каждой конкретной модели может быть различной, причем это зависит не только от желания разработчика, но и от соблюдения им технологии моделирования. А любая технология предполагает наличие определенной последовательности действий.

Компьютерное моделирование - это математическое моделирование с использованием средств вычислительной техники. Соответственно, технология компьютерного моделирования предполагает выполнение следующих действий:

1) определение цели моделирования;
2) разработка концептуальной модели;
3) формализация модели;
4) программная реализация модели;
5) планирование модельных экспериментов;
6) реализация плана эксперимента;
7) анализ и интерпретация результатов моделирования.

Содержание первых двух этапов практически не зависит от математического метода, положенного в основу моделирования (и даже наоборот – их результат определяет выбор метода). А вот реализация остальных шести существенно различается для каждого из двух основных подходов к построению модели. Именуются эти подходы в разных книгах по – разному, мы используем для их обозначения термины «аналитическое» и «имитационное» моделирование.

Аналитическое моделирование предполагает использование математической модели реального объекта в форме алгебраических, дифференциальных, интегральных и других уравнений, связывающих выходные переменные с входными, дополненных системой ограничений. При этом предполагается наличие однозначной вычислительной процедуры получения точного решения уравнений.

При ***имитационном моделировании*** используемая математическая модель воспроизводит логику («алгоритм») функционирования исследуемой системы во времени при различных сочетаниях значений параметров системы и внешней среды.

Примером простейшей аналитической модели может служить уже упоминавшееся уравнение прямолинейного движения. При исследовании такого процесса с помощью имитационной модели должно быть реализовано наблюдение за изменением пройденного пути с течением времени.

Очевидно, в одних случаях более предпочтительным является аналитическое моделирование, в других – имитационное (или сочетание того и другого). Чтобы выбор был удачным, необходимо ответить на два вопроса:

- с какой целью проводится моделирование;
- к какому классу может быть отнесено моделируемое явление.

Ответы на оба эти вопроса могут быть получены в ходе выполнения двух первых этапов моделирования.

Концептуальная модель

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:

1) определение типа системы;
2) описание рабочей нагрузки;
3) декомпозиция 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, в тех случаях, когда 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).

Наиболее важные в этом отношении ***классификационные признаки*** приведены ниже.

1. множество 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.

Различают *два основных типа динамических* систем:

- с дискретными состояниями (множество состояний конечно или счетно);
- с непрерывным множеством состояний.

Возможны смешанные случаи.

2. ***движением системы*** -- процесс смены состояний.

Смена состояний может происходить либо в фиксированные моменты времени, множество которых дискретно и заранее определено (например, поступление новых партий товара на склад), либо непрерывно (изменение курсов валюты в ходе торгов). При этом 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 системы можно однозначно указать, в какое именно состояние она перейдет.

Для *стохастической* системы можно указать лишь множество возможных состояний перехода и, в некоторых случаях, вероятности 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, знание классификационных признаков дает возможность оценить степень ее соответствия первоначальному замыслу разработчика.

3. ***рабочая нагрузка*** – это совокупность внешних воздействий, оказывающих влияние на эффективность применения данной системы в рамках решаемой задачи,

Описание рабочей нагрузки является не только важной, но и достаточно сложной задачей. Особенно в тех случаях, когда приходится учитывать влияние случайных факторов, или когда идет о рабочей проектируемой принципиальной новой системы. В связи с этим многие вводят понятие модели рабочей нагрузки, подчеркивая сопоставимость уровня сложности описания собственно системы и ее рабочей нагрузки.

Модель рабочей нагрузки (РН) должна обладать следующими основными свойствами:

- совместимостью с моделью системы;
- представительностью;
- управляемостью;
- системной независимостью.

Свойство ***совместимости*** предполагает, что, во – первых степень детализации описания РН соответствует детализации описания системы; во – первых, модель РН должна быть сформулирована в тех же категориях предметной области, что и модель системы. Например, если в модели системы исследуется использование ресурсов, РН должна быть выражена в запросах на ресурсы;

Представительность модели РН определяется ее способностью адекватно представить РН в соответствии с целями исследования. Другими словами, модель РН должна отвечать целям исследования системы. Например, если оценивается пропускная способность, должна выбирать РН, «насыщающая» систему.

Под ***управляемостью*** понимается возможность изменения параметров модели РН в некотором диапазоне, определяемом целями исследования.

Системная независимость – это возможность переноса модели РН с одной системы на другую с сохранением ее представительности. Данное свойство наиболее важно при решении задачи сравнения различных систем или различных модификаций одной системы. Если модель РН зависит от конфигурации исследуемой системы или других ее параметров, то использование такой модели для решения задачи выбора невозможно.

И наконец, обратимся к этапу, завершающему построение концептуальной модели системы – ее декомпозиции.

4. 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 проведение.

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 мех 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).

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 эксперимент.

2. МАТЕМАТИЧЕСКИЕ МОДЕЛИ

Под математическим моделированием понимают способ исследования различных процессов путем изучения явлений, имеющих *различное физическое содержание,* но описываемых *одинаковыми математическими соотношениями.* При изучении любого процесса методом математического моделирования необходимо в первую очередь построить его математическое описание, или, как мы далее будем говорить, *математическую модель.* В простейших случаях, математическая модель позволяет для данного процесса-оригинала подобрать на основании известных аналогий удобные физические процессы-модели, а также установить соотношения подобия, связывающие их параметры, без которых трудно использовать результаты моделирования для изучения процесса-оригинала. В более сложных случаях, когда для моделирования создаются специальные моделирующие установки (стенды) или используются вычислительные машины, математическая модель необходима для определения структуры и параметров стенда или построения моделирующего алгоритма.

2.1. Математические модели

Математическая модель, описывает *формализованный* процесс функционирования системы и в состоянии охватить только основные, характерные его закономерности.

Процесс функционирования любой системы будем рассматривать как последовательную смену ее состояний в некотором интервале времени (t_0,t_1). Состояния системы в каждый момент времени t из упомянутого интервала характеризуются набором величин $z_1, z_2, \ldots, z_n$. Процесс функционирования системы рассматриваем как последовательную смену состояний, и $z_1(t), z_2(t), \ldots, z_n(t)$ являются функциями времени t. В дальнейшем будем называть их *характеристиками состояний* системы.

Под *математической моделью* реальной системы будем понимать *совокупность соотношений* (например, формул, уравнений, неравенств, логических условий, операторов и т. д.), *определяющих характеристики состояний системы (а через них и выходные сигналы) в зависимости от параметров системы, входных сигналов, начальных условий и времени.*

Однако при исследовании реальных систем не всегда удается построить математические модели в виде явных функций или уравнений.

Перейдем к некоторым общим замечаниям, связанным с понятием математической модели.

1. *Однозначность* определения характеристик состояний системы и выходных сигналов через параметры системы, входные сигналы и начальные условия. Это требование выполняется для так называемых детерминированных моделей, представляющих собой совокупность

неслучайных соотношений. Если при этом начальные условия и входные сигналы не случайны, то модель оказывается *вполне детерминированной.* На практике нередко приходится рассматривать *случайные* процессы функционирования различных систем. Характеристики состояний системы для таких процессов оказываются случайными функциями времени. Будем говорить, что при помощи математической модели *однозначно* определяются *распределения вероятностей* для характеристик состояний системы, если заданы распределения вероятностей для начальных условий, параметров системы и возмущений, действующих на ее элементы, а также для входных сигналов.

2. *Выбор совокупности параметров*, характеризующих исследуемую систему. Реальные процессы, если их рассматривать во всех деталях, весьма сложны. Учет большого количества второстепенных деталей оказывается практически нецелесообразным. В большинстве случаев при решении прикладных задач достаточно учитывать лишь основные стороны исследуемого процесса. Поэтому обычно при построении математической модели процесса ограничиваются сравнительно небольшим количеством параметров. В таких условиях, естественно, об однозначности определения набора параметров, характеризующих систему, не может быть и речи.
3. *Определение совокупности начальных условий*. На этапе формализации процесса, когда контуры математической модели еще недостаточно выяснены, определить перечень начальных условии не представляется возможным. Когда же математическая модель построена, перечень начальных условий может быть определен однозначно. Естественно, что перечень начальных условий зависит от того, какие выбраны характеристики состояний системы.

Математическая модель может появиться только как следствие четкого формального описания рассматриваемого процесса с требуемой степенью приближения к действительности, только в результате *формализации* процесса.

2.2. Формализация процессов функционирования сложных систем

Математическая модель является результатом *формализации* процесса, т. е. построения четкого формального (математического) описания процесса с необходимой степенью приближения к действительности.

Модель объекта моделирования, т. е. системы S, можно представить в виде множества величин, описывающих процесс функционирования реальной системы и образующих в общем случае следующие подмножества:

- совокупность ***входных воздействий*** на систему $x \in X,\ i=1,..n_x$;
- совокупность ***воздействий внешней среды*** $v_i \in V,\ i=1,\ ..,n_v$;
- совокупность ***внутренних (собственных) параметров*** системы $h_i \in H$,

$i=1, .., n_h$;

- совокупность ***выходных характеристик*** системы $y_i \in Y$, $i=1, .., n_y$;

Причем в перечисленных подмножествах можно выделить управляемые и неуправляемые переменные.

Совокупность зависимостей выходных характеристик системы от времени $y_j(t)$ для всех видов $j=1,\dots, n_y$ называется ***выходной траекторией*** $\bar{y}(t)$. Зависимость называется *законом функционирования системы S* и обозначается *Fs*.

Весьма важным для описания и исследования системы S является понятие ***алгоритма функционирования As***, под которым понимается метод получения выходных характеристик с учетом входных воздействий $x(t)$, воздействий внешней среды $v(t)$ и собственных параметров системы $h\ (t)$. Очевидно, что один и тот же закон функционирования *Fs* системы S может быть реализован различными способами, т. е. с помощью множества различных алгоритмов функционирования *As*.

Очевидно, что детерминированная модель является частным случаем стохастической модели.

Приведенные математические соотношения представляют собой математические схемы общего вида и позволяют описать широкий класс систем. Однако в практике моделирования на первоначальных этапах исследования системы рациональнее использовать ***типовые математические схемы***: дифференциальные уравнения, конечные и вероятностные автоматы, системы массового обслуживания и т. д.

При построении математических моделей процессов функционирования систем можно выделить следующие основные подходы: непрерывно–детерминированный (например, дифференциальные уравнения); дискретно–детерминированный (конечные автоматы); дискретно-стохастический (вероятностные автоматы); непрерывно-стохастический (системы массового обслуживания); обобщенный или универсальный (агрегативные системы).

Математические схемы, рассматриваемые в последующих параграфах данной главы, должны помочь оперировать различными подходами в практической работе при моделировании конкретных систем.

2.3.Математические схемы

2.3.1.Непрерывно-детерминированные модели (D-схемы)

Обычно в таких математических моделях в качестве независимой переменной, от которой зависят неизвестные искомые функции, служит время *t*. Тогда математическое соотношение для детерминированных систем в общем виде будет

$$\bar{y}' = \vec{f}(\bar{y}, t); \bar{y}(t_0) = \bar{y}_0,$$

где:

$\vec{y}' = \dfrac{d\vec{y}}{dt}$, $\vec{y} = (y_1, y_2, ..., y_n)$ и $\vec{f} = (f_1, f_{2,}, ..., f_n)$ -- n-мерные векторы,

$\vec{f}(\vec{y}, t)$ -- вектор-функция, которая определена на некотором (n+1)— мерном множестве (ŷ,t) и является непрерывной.

Так как математические схемы такого вида отражают динамику изучаемой системы, т. е. ее поведение во времени, то они называются ***D-схемами*** (англ. dynamic**).**

Использование *D-схем* позволяет формализовать процесс функционирования непрерывно–детерминированных систем S и оценить их основные характеристики, применяя аналитический или имитационный подход, реализованный в виде соответствующего языка для моделирования непрерывных систем или использующий аналоговые и гибридные средства вычислительной техники.

2.3.2. Дискретно-детерминированные модели (f-схемы)

Особенности дискретно–детерминированного подхода на этапе формализации процесса функционирования систем может быть рассмотрен на примере использования в качестве математического аппарата теории автоматов. На основе этой теории система представляется в виде автомата, перерабатывающего дискретную информацию и меняющего свои внутренние состояния лишь в допустимые моменты времени.

Абстрактно конечный автомат (англ. finite automata) можно представить как математическую схему ***(F-схему),*** характеризующуюся шестью элементами:

- конечным множеством X входных сигналов (входным алфавитом);
- конечным множеством Y выходных сигналов (выходным алфавитом);
- конечным множеством Z внутренних состояний (внутренним алфавитом или алфавитом состояний);
- начальным состоянием z_0, $z_0 \in Z$;
- функцией переходов $\varphi(z,x)$;
- функцией выходов $\psi(z, x)$.

Автомат, задаваемый *F-схемой,* принято обозначать*:*

$$F=<\boldsymbol{Z}, \boldsymbol{X}, \boldsymbol{Y}, \varphi, \psi, z_0>.$$

Абстрактный конечный автомат имеет один входной и один выходной каналы. В каждый момент *t=0,* 1, 2,... дискретного времени F-автомат находится в определенном состоянии *z(t)* из множества ***Z*** состояний автомата, причем в начальный момент времени *i=0* он всегда находится в начальном состоянии $z(0)=z_o$. В момент *t,* будучи в состоянии *z(t),* автомат способен воспринять на входном канале сигнал $x(t) \in X$ и выдать на выходном канале сигнал $y(t)=\psi[z(t),\ x(t)]$, переходя в состояние $z(t+1)= =\varphi[z(t),x(t)]$, $z(t) \in \boldsymbol{Z}$, $y(t) \in \boldsymbol{Y}$. Другими словами, если на вход конечного автомата, установленного в начальное состояние z_0, подавать в некоторой последовательности буквы входного алфавита *x(0), x(1), x(2),...,* т. е. входное слово, то на выходе автомата будут последовательно появляться буквы выходного алфавита $y(0)$, $y(1)$, y (2),..., образуя выходное слово.

Сказанное выше можно описать следующими уравнениями: для F-автомата первого рода, называемого также автоматом Мили,

$$z(t+1)= =\varphi[z(t),x(t)],\ y(t)=\psi[z(t),\ x(t)],\ t=0,1,2,...; \qquad (2.1)$$

для F-автомата второго рода

$$z(t+1)=\varphi[z(t),x(t)],\ y(t)=\psi[z(t),\ x(t\text{-}1)],\ t=0,1,2,...; \qquad (2.2)$$

Автомат второго рода, для которого $y(t)=\psi[z(t)]$, t=0,1,2,..., т. е. функция выходов не зависит от входной переменной x $\{t)$, называется автоматом Мура.

По числу состояний различают конечные автоматы с памятью и без памяти. Автоматы с памятью имеют более одного состояния, а автоматы без памяти (комбинационные или логические схемы) обладают лишь одним состоянием.

По характеру отсчета дискретного времени конечные автоматы делятся на *синхронные и асинхронные*. В синхронных F-автоматах моменты времени, в которые автомат «считывает» входные сигналы, определяются принудительно синхронизирующими сигналами. Реакция автомата на каждое значение входного сигнала заканчивается за один такт, длительность которого определяется интервалом между соседними синхронизирующими сигналами. Асинхронный F-автомат считывает входной сигнал непрерывно и поэтому, реагируя на достаточно длинный входной сигнал постоянной величины *x*, он может, несколько раз изменять состояние, выдавая соответствующее число выходных сигналов, пока не перейдет в устойчивое, которое уже не может быть изменено данным входным.

2.3.3. Дискретно-стохастические модели (Р-схемы)

Рассмотрим особенности построения математических схем при дискретно-стохастическом подходе к формализации процесса функционирования исследуемой системы S. Поскольку сущность дискретизации времени при этом подходе остается аналогичной рассмотренным конечным автоматам, то влияние фактора стохастичности проследим также на разновидности таких автоматов, а именно на вероятностных (стохастических) автоматах. В общем виде *вероятностный автомат* (англ. probabilistic automat) 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 статистически.

Вероятностный автомат может быть описан либо таблицей переходов, либо матрицей переходов Р-автомата и начальным распределением вероятностей. Математический аппарат, используемый при исследовании Р-автоматов, является аппарат марковских цепей.

2.3.4. Непрерывно-стохастические модели (q-схемы)

Особенности непрерывно-стохастического подхода рассмотрим на примере использования в качестве типовых математических схем *систем массового обслуживания,* (англ. queueing system), которые будем называть *Q-схемами.* Системы массового обслуживания представляют собой класс математических схем, разработанных в теории массового обслуживания и различных приложениях для формализации процессов функционирования систем, которые по своей сути являются процессами обслуживания.

В качестве процесса обслуживания могут быть представлены различные по своей физической природе процессы функционирования экономических, производственных, технических и других систем, например: потоки поставок продукции некоторому предприятию, потоки деталей и комплектующих изделий на сборочном конвейере цеха, заявки на обработку информации ЭВМ от удаленных терминалов и т. д. При этом характерным для работы таких объектов является случайное появление заявок (требований) на обслуживание и завершение обслуживания в случайные моменты времени, т. е. стохастический характер процесса их функционирования. Остановимся на основных понятиях массового обслуживания, необходимых для использования *Q-схем* как при аналитическом, так и при имитационном подходе.

Работа любой системы массового обслуживания состоит в выполнении поступающего на ее вход потока заявок. Заявки поступают в некоторые, в общем случае случайные, моменты времени. Обслуживание заявки продолжается какое-то время, также случайное, после чего канал освобождается для обслуживания следующей заявки. Предмет теории массового обслуживания – установление зависимостей между характером потока заявок, производительностью отдельного канала обслуживания, числом каналов и эффективностью обслуживания.

Случайный процесс, протекающий в СМО состоит в том, что система в случайные моменты времени переходит из одного состояния в другое. СМО представляет собой физическую систему дискретного типа, а переход системы из одного состояния в другое происходит скачком.

В любой момент времени система пребывает в одном из возможных состояний и очевидно, что для любого t справедливо:

$$\sum_k p_k(t) = 1$$

Случайные процессы со счетным множеством состояний бывают двух типов: с дискретным или непрерывным временем.

С дискретным временем: переходы из состояния в состояние могут происходить только в строго определенные, разделенные конечными интервалами моменты времени t_1, t_2 … .

С непрерывным временем: переход системы из состояния в состояние возможен в любой момент времени.

Случайные процессы, протекающие в СМО как правило являются процессами с непрерывным временем. Граф перехода системы из состояние в состояние может быть проиллюстрирован рис.

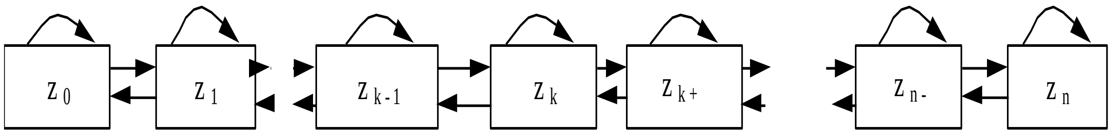


Рис. 2.6

2.3.5. Обобщенные модели (А-схемы)

Наиболее известным общим подходом к формальному описанию процессов функционирования систем является подход, предложенный Н.П.Бусленко. Этот подход позволяет описывать поведение непрерывных и дискретных, детерминированных и стохастических, т.е. по сравнению с рассмотренными является обобщенным (универсальным) и базируется на понятии агрегатируемой системы, представляющей собой формальную схему общего вида, которую принято называть А-схемой.

Анализ существующих средств моделирования показывает, что комплексное решение проблем, возникающих в процессе создания и реализации модели, возможно только в том случае, когда моделирующие системы имеют в своей основе единую формальную математическую схему. Такая схема должна выполнять следующие функции:

- являться адекватным математическим описанием объекта моделирования;
- служить основой для построения алгоритмов и программ, реализующих модель;
- позволять в упрощенном варианте проводить аналитические исследования.

В качестве элемента А-схемы выступает агрегат. Связь между агрегатами (внутри системы S и внешней средой E) осуществляется опеатором R. Агрегат может разбиваться на агрегаты следующего уровня.

Любой агрегат характеризуется следующими множествами:

- моментами времени T;
- входными сигналами X;
- выходными сигналами У;
- состояниями на каждый момент времени Z(t).

Переход агрегата из состояния в состояние происходит за малый интервал времени δt $z(t_2) \neq z(t_1)$. Изменение состояния определяется скачком δz. Агрегат из состояния в состояние переходит в зависимости от собственных (внутренних) параметров h(t) и входных сигналов x(t).

В начальный момент времени агрегат находится в состоянии $z(t_0)=z_0$, которое задается законом $L(z(t_0))$.

Процесс функционирования агрегата в случае воздействия сигнала x_n описывается случайным оператором V. Пусть в момент времени t_n поступил сигнал x_n. Состояние агрегата определиться так:

$Z(t_n+0)=V(t_n,z(t_n),x_n)$.

Если в течение времени (t_n,t_{n+1}) не пришло ин одного входного сигнала, то агрегат может перейти в другое состояние за счет изменение внутреннего состояния в соответствии со случайным оператором U:

$z(t)=U(t,t_n,z(t_n+0))$.

Совокупность случайных операторов V и U рассматривается как оператор перехода автомата в новые состояния. При этом процесс функционирования агрегата состоит из скачков состояний δz в моменты поступления новых сигналов x и изменений состояний агрегата между этими моментами. Моменты скачков δz называются особыми состояниями А-схемы. Для описания скачков в особые моменты используется оператор W, представляющий собой частный случай оператора U:

$z(t_\delta)=W(t_\delta,z(t_\delta))$.

В множестве состояний агрегата выделяется подмножество $Z^{(Y)}$, которое является подмножеством выдачи выходного сигнала:

$Y=G(t_\delta,z(t_\delta))$.

Таким образом, под агрегатом будем понимать объект, определяемый упорядоченной совокупностью рассмотренных множеств T, X, Y, Z, $Z^{(Y)}$, H и случайных операторов V, U, W, G.

Последовательность входных сигналов, расположенных в порядке поступления их на вход А-схемы называют входным сообщением, а последовательных выходных – выходным сообщением.

Существует класс больших систем, которые ввиду их сложности не могут быть формализованы в виде математических схем одиночных агрегатов, поэтому их формализуют некоторой конструкцией из отдельных агрегатов. Для описания системы в целом, необходимо иметь описание как отдельных агрегатов, так и связей между ними.

Для построения формального понятия А-схемы необходимо выбрать способы математического описания взаимодействия между агрегатами. Для этого вводится ряд предположений о закономерностях функционирования А-схем, которые согласуются с опытом исследования реальных сложных систем:

- взаимодействие между А-схемой и внешней средой E, а также между отдельными агрегатами внутри системы осуществляется при передаче сигналов, причем взаимные влияния, имеющие место вне механизма передачи сигналов не учитываются;
- для описания сигнала достаточно некоторого конечного набора характеристик;
- элементарные сигналы мгновенно передаются в А-схеме независимо друг от друга по элементарным каналам;
- ко входному контакту любого элемента А-схемы подключается не более чем один элементарный канал, к выходному контакту – любое конечное число элементарных каналов.

Взаимодействие А-схемы с внешней средой рассматривается как обмен сигналами между внешней средой и элементами А-схемы. В связи с этим внешнюю среду можно представить в виде фиктивного элемента А-схемы.

Таким образом, использование обобщенной типовой математической схемы моделирования А_схемы в принципе не отличается от использования рассмотренных ранее D, F, P, Q-схем. Для частного случая результаты могут быть получены аналитическим методом. В более сложных случаях прибегают к имитационному методу.

Представление объекта моделирования в виде А-схемы может являться тем фундаментом, на котором базируется построение имитационной системы и ее внешнего и внутреннего математического обеспечения. Стандартная форма представления исследуемого объекта в виде А-схемы приводит к унификации не только алгоритмов имитации, но и к возможности применять стандартные методы обработки и анализа результатов моделирования.

3. МОДЕЛИРОВАНИЕ СИСТЕМ МАССОВОГО ОБСЛУЖИВАНИЯ

3.1. Понятие СМО

Системой массового обслуживания (СМО) называется любая система предназначенная для обслуживания каких-либо заявок (требований), поступающих на нее в случайные моменты времени.

В качестве процесса обслуживания могут быть представлены различные по своей физической природе процессы функционирования экономических, производственных, технических и других систем. Примеры систем массового обслуживания следующие: потоки поставок продукции некоторому предприятию, потоки деталей и комплектующих изделий на сборочном конвейере цеха, заявки на обработку информации ЭВМ от удаленных терминалов и т. д. При этом характерным для работы таких объектов является случайное появление заявок (требований) на обслуживание и завершение обслуживания в случайные моменты времени, т. е. стохастический характер процесса их функционирования. Остановимся на основных понятиях массового обслуживания, необходимых как при аналитическом, так и при имитационном подходе.

Работа любой системы массового обслуживания состоит в выполнении поступающего на ее вход потока ***заявок***. Заявки поступают в некоторые, в общем случае случайные, моменты времени. Обслуживание заявки продолжается какое-то время, также случайное, после чего канал освобождается для обслуживания следующей заявки. Предмет теории массового обслуживания – установление зависимостей между характером потока заявок, производительностью отдельного канала обслуживания, числом каналов и эффективностью обслуживания.

Различают СМО *с отказами и СМО с очередью.* В СМО с отказами заявка, пришедшая в момент, когда все каналы заняты, получает отказ, покидает СМО и в дальнейшем в процессе ее работы не участвует. В СМО с очередью заявка, пришедшая в момент занятости всех каналов, не покидает СМО, а становится в очередь и ждет, пока не освободится какой-нибудь канал. Число мест в очереди m может быть как ограниченным, так и неограниченным. При m = О СМО с очередью превращается в СМО с отказами. Очередь может иметь ограничения не только по количеству стоящих в ней заявок (длине очереди), но и по времени ожидания (такие СМО называются «системами с нетерпеливыми клиентами»).

СМО с очередью различаются не только по ограничениям очереди, но и по *дисциплине обслуживания:* обслуживаются ли заявки в порядке поступления, или в случайном порядке, или же некоторые заявки обслуживаются вне очереди (так называемые «СМО с приоритетом»). Приоритет может иметь несколько градаций или рангов.

Аналитическое исследование СМО является наиболее простым, если все потоки событий, переводящие ее из состояния в состояние, — простейшие (стационарные пуассоновские). Это значит, что интервалы времени между событиями в потоках имеют показательное распределение с параметром, равным интенсивности соответствующего потока. Для СМО это допущение означает, что как поток заявок, так и поток обслуживании — простейшие. Под *потоком обслуживании* понимается поток заявок, обслуживаемых одна за другой одним непрерывно занятым каналом. Этот поток оказывается простейшим, только если время обслуживания заявки $T_{обс}$ представляет собой случайную величину, имеющую показательное распределение. Параметр этого распределения μ есть величина, обратная среднему времени обслуживания. Вместо «поток обслуживании — простейший» часто говорят «время обслуживания — показательное». Условимся в дальнейшем для краткости всякую СМО, в которой все потоки простейшие, называть *простейшей* СМО. В этой главе мы будет рассматривать главным образом простейшие СМО.

Если всё потоки событий простейшие, то процесс, протекающий в СМО, представляет собой марковский случайный процесс с дискретными состояниями и непрерывным, временем. При выполнении некоторых условий для этого процесса существует финальный стационарный режим, при котором как вероятности состояний, так и другие характеристики процесса не зависят от времени.

Задачи теории массового обслуживания — нахождение вероятностей различных состояний СМО, а также установление зависимости между заданными параметрами (числом каналов n, интенсивностью потока заявок λ, распределением времени обслуживания и т, д.) и *характеристиками эффективности* работы СМО. В качестве таких характеристик могут рассматриваться, например, следующие:

- среднее число заявок *A,* обслуживаемое СМО в единицу времени, или *абсолютная пропускная способность* СМО;
- вероятность обслуживания- поступившей заявки Q или *относительная пропускная способность* СМО; $Q = A/\lambda$;
- вероятность отказа $P_{отк}$ т.е вероятность того, что поступившая заявка не будет обслужена, получит отказ; $P_{отк} = 1 - Q$;

Рассмотри процессы, протекающие в системе массового обслуживания.

3.2.Мнемоническое обозначение СМО.

В теории массового обслуживания приняты очень удобные сокращенные обозначения для различных СМО, позволяющие легко охарактеризовать систему. В основе этих обозначений лежит трехбуквенная комбинация вида A/B/N, где:

A — описывает распределение (или задает характер закона распределения) интервалов поступления заявок;

B — описывает распределение длительностей обслуживания заявок;

N — задает количество обслуживающих приборов в СМО.

Иногда, когда СМО является системой с ограниченной емкостью накопителя (или с ограниченной очередью), приведенное обозначение расширяется до четырех букв A/B/N/К, где последняя буква (на самом деле число, как и N) K задает емкость накопителя (количество мест ожидания).

Приведенные трех или четырех буквенные обозначения называют обозначениями Кендалла. В этих обозначениях A и B могут принимать значения из следующего набора символов {M, D, Ek, Hk, G, U}. При этом:

а) A или B=M, если распределение интервалов поступления или длительностей обслуживания заявок является экспоненциальным (M — от слова Markovian — Марковский);

б) A или B=D, если интервалы поступления или длительности обслуживания являются детерминированными (D — Determinate);

в) A или B=Ek, если соответствующие распределения являются Эрланговскими порядка k (E — Erlang);

г) A или B=Hk, в случае гиперэкспоненциальных распределений порядка k (H — Hyperexponential);

д) A или B= G, в случае распределений общего (произвольного) вида (G — General — общий, общего вида);

е) A или B= U — при равномерных распределениях соответствующих случайных величин (U — Uniform distribution — равномерное распределение).

Так, например, обозначение вида:

M/M/1 означает СМО с простейшим потоком на входе и экспоненциально распределенной длительностью обслуживания заявок в приборе (один)

D/E2/3/5 — СМО с регулярным потоком на входе, длительностью обслуживания, распределенной по закону Эрланга 2-го порядка, тремя обслуживающими приборами и пятью местами ожидания;

M/G/2 — СМО с простейшим потоком на входе, длительностью обслуживания, распределенная по закону произвольного вида, и двумя обслуживающими приборами.

В случае СМО с неоднородной нагрузкой используются обозначения вида, где символ вектора над буквами A и B указывает на неоднородность нагрузки, а индекс H задает количество классов заявок. Например, — это обозначение СМО с одним обслуживающим прибором, четырьмя классами заявок, которые образуют на входе системы простейшие потоки и имеют общие законы распределения длительностей обслуживания.

3.3. СМО с отказами

Системы массового обслуживания делятся на системы с отказами и системы с ожиданием.

В системах с отказами заявка, поступившая в момент, когда все каналы обслуживания заняты, немедленно получает отказ, покидает систему и в дальнейшем в процессе обслуживания не участвует.

Пусть имеется n-канальная СМО с отказами. Рассмотрим конечное множество состояний этой системы:

z_0 – свободны все каналы;

z_1 – занят один канал;

. . . .

z_k – заняты k каналов;

. . . .

z_n – заняты все n каналов.

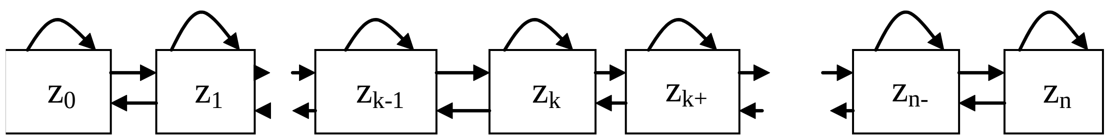


Рис. 3.1

Определим вероятности состояния системы pk(t) для любого момента времени в предположении, что поток заявок простейший, с интенсивностью λ, время обслуживания показательное, с параметром μ.

Поскольку оба потока заявок в системе (заявок и обслуживания) являются простейшими, то процесс, протекающий в системе будет марковским.

Очевидно, что для любого момента времени

$$\sum_{k=0}^{n} p_k(t) = 1$$

.

Составим дифференциальные уравнения для всех вероятностей состояний системы. Для этого, зафиксируем момент времени t и найдем вероятность pk(t+Δt) того, что в момент (t+Δt) система будет находиться в состоянии zk.

Для состояния z0 это может произойти двумя способами:

событие А – в момент времени t система находилась в состоянии z0 и осталась в этом состоянии. Вероятность этого события равна вероятности того, что за время Δt на вход системы не пришла ни одной заявки:

$$e^{-\lambda \cdot \Delta t} \approx 1 - \lambda \cdot \Delta t.$$

Следовательно, P(A)=p0(t)(1-λ˙Δt).

событие В – вероятность того, что система была в состоянии z1 и перешла в состояние z0. Вероятность этого события равна:

$$1 - e^{-\mu \cdot \Delta t} \approx \mu \cdot \Delta t.$$

Следовательно, P(B)=p1(t)μ˙Δt.

Таким образом:

p0(t+Δt)=p0(t)(1-λ˙Δt)+ p1(t)μ˙Δt.

$$\frac{dp_0(t)}{dt} = -\lambda p_0(t) + \mu p_1(t).$$

Аналогично составляются дифференциальные уравнения для других состояний системы. Для состояния zk вероятность pk(t+Δt) определиться как сумма вероятностей трех событий:

событие А – в момент времени t система находилась в состоянии zk и осталась в этом состоянии. Вероятность этого события равна вероятности того, что за время Δt на вход системы не пришла ни одной заявки и ни одна из k заявок из системы не ушла (не обслужилась):

$$e^{-\lambda\cdot\Delta t}(e^{-\mu\Delta t})^k = e^{-(\lambda+k\mu)\Delta t} \approx 1-(\lambda+k\mu)\cdot\Delta t.$$

Следовательно, P(A)=pk(t)[1-(λ+kμμ)˙Δt].

событие В – вероятность того, что система была в состоянии zk-1 и перешла в состояние zk. (пришла одна заявка). Вероятность этого события равна:

P(B)=pk-1(t)λ˙Δt.

событие С – вероятность того, что система была в состоянии zk+1 и перешла в состояние zk. (обслужена одна заявка). Вероятность этого события равна:

P(C)=pk+1(t)(k+1)μ˙Δt.

Таким образом:

pk(t+Δt)= pk(t)[1-(λ+kμμ)˙Δt]+ pk-1(t)λ˙Δt+ pk+1(t)(k+1)μ˙Δt.

$$\frac{dp_k(t)}{dt} = \lambda p_{k-1}(t) - (\lambda + k\mu)\, p_k(t) + (k+1)\mu p_{k+1}(t).$$

Составим уравнение для последней вероятности pn:

pn(t+Δt)≈ pn(t)(1-nμ˙Δt)+ pn-1(t)λ˙Δt.

$$\frac{dp_n(t)}{dt} = \lambda p_n(t) - n\mu p_{n-1}(t).$$

Таким образом, получена система дифференциальных уравнений для вероятностей состояний системы:

$$\begin{cases} \dfrac{dp_0(t)}{dt} = -\lambda p_0(t) + \mu p_1(t), \\ \ldots\ldots\ldots \\ \dfrac{dp_k(t)}{dt} = \lambda p_{k-1}(t) - (\lambda + k\mu)\, p_k(t) + (k+1)\mu p_{k+1}(t). \\ \ldots\ldots \qquad\qquad (0<k<n), \\ \dfrac{dp_n(t)}{dt} = \lambda p_n(t) - n\mu p_{n-1}(t). \end{cases}$$

Эти уравнения называются уравнениями Эрланга.

Вероятности $p_k(t)$ характеризуют среднюю загрузку системы и ее изменение с течением времени.

Вероятность $p_n(t)=P_{отк}$ есть вероятность того, что заявка, пришедшая в систему в момент времени t получит отказ.

Величина $q(t)=1-p_n(t)$ называется пропускной способностью системы.

Введем обозначение $\alpha=\lambda/\mu$ и назовем величину α *приведенной плотностью потока заявок*. Эта величина есть также среднее число заявок, приходящееся на среднее время обслуживания одной заявки: $\alpha=\lambda m_{обсл}$.

В новых обозначениях вероятности p_k принимает вид:

$$p_k = \frac{\alpha^k}{k!} p_0.$$

Приведенные выше формулы выражают вероятности p_k через p_0. Для того, чтобы выразить эти вероятности через характеристики системы α и n, воспользуемся условием нормировки:

$$\sum_{k=0}^{n} p_k = p_0 \sum_{k=0}^{n} \frac{\alpha^k}{k!} = 1,$$

откуда

$$p_0 = \frac{1}{\sum_{k=0}^{n} \frac{\alpha^k}{k!}}.$$

Окончательное выражение для вероятностей состояния системы принимают вид:

$$p_k = \frac{\frac{a^k}{k!}}{\sum_{k=0}^{n} \frac{\alpha^k}{k!}} \qquad (0 \le k \le n).$$

Вероятность отказа (все каналы заняты):

$$P_{отк} = p_n = \frac{\frac{\alpha^n}{n!}}{\sum_{k=0}^{n} \frac{a^k}{k!}}.$$

Для одноканальной системы (n=1):

$$P_{отк} = p_1 = \frac{\alpha}{1+\alpha}.$$

Относительная пропускная способность :

$$q = 1 - P_{отк} = \frac{1}{1+\alpha}.$$

Формулы Эрланга и их следствия были получены в предположении о показательном распределении времени обслуживания заявок. Однако исследования показали, что эти формулы справедливы при любом законе распределения времени обслуживания, лишь бы входной поток был простейшим.

3.4. СМО с ожиданием

Система массового обслуживания называется системой с ожиданием, если заявка, заставшая все каналы занятыми, становится в очередь и ждет, пока не освободится какой-нибудь канал.

Если время ожидания заявки в очереди ничем не ограничено, то система называется *чистой системой с ожиданием*. Если оно ограничено некоторыми условиями, то система называется системой смешанного типа. Ограничения, наложенные на ожидание могут быть различного типа, например:

- ограничение на время пребывания заявки в очереди;
- ограничение на длину очереди;
- ограничение на время пребывания заявки в системе.

В системах с ожиданием существенную роль играет так называемая *дисциплина очереди*. Каждый тип системы с ожиданием имеет свои особенности и математическую теорию. Мы остановимся на простейшем случае смешанной системы, являющимся обобщением задачи Эрланга для системы с отказами.

Рассмотрим СМО с n каналами, на вход которой поступает простейший поток с параметром λ. Время обслуживания заявок также имеет показательное распределение с параметром μ. Заявка, заставшая все каналы занятыми, становится в очередь и ожидает обслуживания. Время ожидания заявки в очереди ограничено некоторым сроком $T_{ож}$. Если до истечения этого срока заявка не будет обслужена, то она покидает систему. Срок ожидания обслуживания будем полагать случайной величиной с показательным распределением и параметром ν. Очевидно, что при $\nu\to\infty$, система смешанного типа превращается в чистую систему с отказами, а при $\nu\to 0$, система смешанного типа превращается в чистую систему с ожиданиями.

Отметим, что в предположении о показательном распределении срока ожидания пропускная способность системы не зависит от того, обслуживаются ли заявки в порядке очереди ли в случайно порядке: для каждой заявки закон распределения оставшегося времени ожидания не зависит от того, сколько времени заявка стояла в очереди.

- для любого $k\le n$ $p_k = \dfrac{\lambda^k}{k!\mu^k} p_0$;
- для любого $s\ge 1$: $p_{n+s} = \dfrac{\lambda^{n+s} p_0}{n!\mu^n \prod\limits_{m=1}^{s}(n\mu + m\nu)}$.

В приведенных выше формулах в качестве сомножителя присутствует вероятность p_0. Определим эту вероятность из дополнительного условия:

$$p_0\left\{\sum_{k=0}^{n}\frac{\lambda^k}{k!\mu^k}+\sum_{s=1}^{\infty}\frac{\lambda^{n+s}}{n!\mu^n\prod_{m=1}^{s}(n\mu+m\nu)}\right\}=1$$

Введем обозначения:

$\lambda/\mu=\lambda m_{t_{обсл}}=\alpha$; $\nu/\mu=\nu m_{t_{обсл}}=\beta$.

Параметра α и β выражают соответственно среднее число заявок и среднее число необслуженных заявок приходящееся на среднее время обслуживания одной заявки.

В новых обозначениях приведенные выше выражения принимают вид:

$$p_k=\frac{\alpha^k}{k!}p_0; \qquad (0<k\le n)$$

$$p_{n+s}=\frac{\frac{\alpha^{n+s}}{n!}p_0}{\prod_{m=1}^{s}(n+m\beta)}; \qquad (s\ge 1).$$

Зная вероятности состояния системы можно определить и другие интересующие нас характеристики, в частности вероятность того, что заявка покинет систему не обслуженной. Определим эту вероятность из следующих соображений: при установившемся режиме вероятность $P_н$ есть отношение среднего числа заявок, уходящих из очереди в единицу времени. Определим среднее число заявок, находящихся в очереди:

$$m_s=\sum_{s=1}^{\infty}sp_{n+s}.$$

Чтобы найти вероятность $P_н$, нужно среднее число заявок в очереди умножить на среднюю плотность уходов (определим среднее число заявок, покидающих систему) и умножим на интенсивность входного потока заявок:

$$P_н=m_s\cdot\frac{\nu}{\lambda}=\frac{\beta}{\alpha}m_s.$$

Относительная пропускная способность системы: q=1-Pн.

Очевидно, что пропускная система с ожиданиями выше, чем пропускная способность системы с отказами и пропускная способность увеличивается с увеличением среднего времени ожидания mtож=1/ν.

Рассмотрим, во что превратиться система с ожиданиями при изменении параметра β. Очевидно, что при β→∞ система с ожиданиями превращается в чистую систему с отказами, а при β→0 – в чистую систему с ожиданиями. В такой системе вероятность того, что заявка уйдет из системы не обслуженной, равна нулю. Однако, в такой системе не всегда имеется предельный стационарный режим при t→∞. Такой режим существует только при α<n, т.е., когда среднее число заявок, приходящееся на время

обслуживания одной заявки не выходит за пределы возможностей n-канальной системы. В противном случае, число заявок в очереди будет неограниченно возрастать.

Полагая, что α<n, найдем предельные вероятности состояния системы (β→0):

$$p_0 = \frac{1}{\sum_{k=0}^{n}\frac{\alpha^k}{k!} + \frac{\alpha^n}{n!}\sum_{s=1}^{\infty}\frac{\alpha^s}{n^s}} = \frac{1}{\sum_{k=0}^{n}\frac{\alpha^k}{k!} + \frac{\alpha^{n+1}}{n!(n-\alpha)}}.$$

Отсюда найдем:

$$p_k = \frac{\frac{\alpha^k}{k!}}{\sum_{k=0}^{n}\frac{\alpha^k}{k!} + \frac{\alpha^{n+1}}{n!(n-\alpha)}} \qquad (0 \le k \le n).$$

$$p_{n+s} = \frac{\frac{\alpha^{n+s}}{n!n^s}}{\sum_{k=0}^{n}\frac{\alpha^k}{k!} + \frac{\alpha^{n+1}}{n!(n-\alpha)}} \qquad (s \ge 0).$$

Среднее число заявок в очереди:

$$m_s = \frac{\frac{\alpha^{n+1}}{n \cdot n!(1-\frac{\alpha}{n})^2}}{\sum_{k=0}^{n}\frac{\alpha^k}{k!} + \frac{\alpha^{n+1}}{n!(n-\alpha)}}.$$

9.8 Простейшая многофазовая СМО с очередью.

Анализ многофазовых СМО в общем случае затруднен, тем что входящий поток каждой последующей фазы является выходным потоком предыдущей и в общем случае имеет последействие. Однако *если на вход С МО с неограниченной очередью поступает простейший поток заявок, а время обслуживания показательное, то выходной поток, этой СМО — простейший,* с той же интенсивностью λ, что и входящий. Из этого следует, что многофазовую СМО с неограниченной очередью перед каждой фазой, простейшим входящим потоком заявок и показательным временем обслуживания на каждой фазе можно анализировать как простую последовательность простейших СМО.

Если очередь к фазе ограничена, то выходной поток этой фазы перестает быть простейшим и вышеуказанный прием может применяться только в качестве приближенного.

4. ПРИНЦИПЫ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ

4.1. Понятие статистического эксперимента

Имитационное моделирование представляет собой наблюдение поведения модели системы под влиянием входных воздействий. При этом часть из них (а может быть и все) носят случайный характер. В результате такого наблюдения исследователь получает набор экспериментальных данных, на основе которых могут быть оценены характеристики системы.

Очевидно, что аналитические модели для проведения имитационного эксперимента не годятся, и здесь нужна специальная «имитационная» модель, которая должна отвечать следующим основным требованиям:

- Отражать логику функционирования исследуемой системы во времени;
- Обеспечить возможность проведения статистического эксперимента.

Одним из основных понятий имитационного моделирования является понятие статистического эксперимента.

В его основе лежит метод статистических испытаний (метод Монте-Карло). Суть метода заключается в том, что результат испытания ставится в зависимость от значения некоторой случайной величины (СВ), распределенной по заданному закону. Результат каждого конкретного испытания носит случайный характер.

Проведя серию испытаний получают множество частных значений наблюдаемой характеристики (то есть выборку). Поученные статистические данные обрабатываются и представляются в виде численных оценок интересующих исследователя параметров.

Отметим, что метод статистических испытаний применим для исследования как стохастических, так и детерминированных систем.

Важной особенностью метода является то, что его применение практически невозможно без использования компьютерной техники.

Имитационное моделирование не ограничивается разработкой модели и написанием соответствующей программы, а требует подготовки и проведения статистического эксперимента. В вязи с этим результаты имитационного моделирования следует рассматривать как экспериментальные данные, требующие специальной обработки и анализа. Для любого модельного эксперимента необходимо ответить на следующие вопросы:

1. Какова должна быть продолжительность эксперимента для достижения стационарных условий?
2. Как получить статистически независимые наблюдения?
3. Сколько наблюдений необходимо для обеспечения требуемой точности?

4.2. Область применения и классификация имитационных моделей

Имитационная модель (ИМ) — это формальное (то есть выполненное на некотором формальном языке) описание логики функционирования исследуемой системы и взаимодействия отдельных ее элементов во времени, учитывающее наиболее существенные причинно-следственные связи, присущие системе, и обеспечивающее проведение статистических экспериментов.

Необходимо отметить два важных обстоятельства:

1) взаимосвязь между отдельными элементами системы, описанными в модели, а также между некоторыми величинами (параметрами) может быть представлена в виде аналитических зависимостей (например, при моделировании полета управляемой ракеты отработка поступающих на борт команд может быть описана на уровне логики, а возникающие перегрузки рассчитываются аналитически);

2) модель можно считать реализуемой и имеющей практическую ценность только в том случае, если в ней отражены лишь те свойства реальной системы, которые влияют на значение выбранного показателя эффективности.

Как было отмечено выше, для ИМ практически отсутствуют ограничения на область их применения (по типу моделируемой системы), и речь может идти только о целесообразности использования ИМ в данной предметной области и об объеме трудозатрат на ее разработку.

Поскольку основой имитационного моделирования является метод статистических испытаний, наибольший эффект от его применения достигается при исследовании сложных систем, на функционирование которых существенное влияние оказывают случайные факторы.

Применение имитационного моделирования целесообразно также в следующих случаях:

1) если не существует законченной постановки задачи на исследование и идет процесс познания объекта моделирования;

2) если характер протекающих в системе процессов не позволяет описать эти процессы в аналитической форме;

3) если необходимо наблюдать за поведением системы (или отдельных ее компонентов) в течение определенного периода, в том числе с изменением скорости протекания процессов;

4) при изучении новых ситуаций в системе либо при оценке функционирования ее в новых условиях;

5) если исследуемая система является элементом более сложной системы, другие элементы которой имеют реальное воплощение;

6) когда необходимо исследовать поведение системы при введении в нее новых компонентов;

7) при подготовке специалистов и освоении новой техники (в качестве тренажеров).

Но имитационные модели имеют целый ряд недостатков. Первый, и весьма существенный, заключается в том, что разработка ИМ, как правило, требует больших затрат времени и сил. Кроме того, любая имитационная модель сложной системы значительно менее «объективна», чем аналитическая модель, поскольку она прежде всего отражает субъективные представления разработчика о моделируемой системе. Причем бывает достаточно сложно как опровергнуть, так и обосновать адекватность созданной ИМ, особенно если речь идет о проектируемой системе. И, наконец, еще одно обстоятельство. Результаты имитационного моделирования, как и при любом численном методе, всегда носят частный характер. Для получения обоснованных выводов необходимо проведение серии модельных экспериментов, а обработка результатов требует применения специальных статистических процедур.

Каким же образом можно преодолеть указанные недостатки?

Во-первых, современное состояние вычислительной техники и ее программного обеспечения позволило создать пакеты моделирования, использование которых существенно сокращает трудозатраты на создание моделей, статистический анализ и визуализацию полученных результатов.

Во-вторых, «объективность» создаваемой модели может быть обеспечена в том случае, когда для каждого варианта постановки задачи исследования выбирается соответствующая схема построения модели.

В этом отношении знание существующих схем построения имитационных моделей является весьма полезным.

Наиболее важный признак — ***способ представления в модели динамики (движения) системы.*** Она может быть описана посредством событий, работ (активностей), процессов и транзактов.

Другой важный признак — ***способ изменения модельного времени.*** По этому признаку различают моделирование с постоянным шагом и моделирование по особым состояниям.

Все эти понятия являются основополагающими в теории имитационного моделирования.

В зависимости от этапа и назначения проводимых исследований применяется один из трех наиболее распространенных видов имитационных экспериментов:

1) исследование относительного влияния различных факторов на значения выходных характеристик системы;

2) нахождение аналитической зависимости между интересующими исследователя выходными характеристиками и факторами;

3) отыскание оптимальных значений параметров системы (так называемый «экстремальный эксперимент»).

Вид эксперимента влияет не только на выбор схемы ее формализации, но также на построение плана эксперимента и выбор метода обработки его результатов.

С точки зрения организации взаимодействия исследователя с моделью в ходе эксперимента ИМ делятся на автоматические и диалоговые.

Автоматическими называются ИМ, взаимодействие пользователя с которыми сводится только к вводу исходной информации и управлению началом и окончанием работы моделей.

Диалоговыми называются ИМ, позволяющие исследователю активно управлять ходом моделирования.

4.3. Описание поведения системы

Описание динамики системы, или, проще говоря, ее поведения, составляет основу любой имитационной модели. В качестве исходных посылок для решения этой задачи используются результаты, полученные на этапе разработки концептуальной модели системы. К ним относятся:

• определение принадлежности моделируемой системы одному из известных классов;

• описание рабочей нагрузки системы;

• выбор уровня детализации представления системы в модели и ее декомпозиция.

Все последующие действия исследователя по созданию модели могут быть отнесены к этапу ее формализации, который в общем случае предполагает:

• выбор метода отображения динамики системы (на основе событий, процессов или транзактов);

• формальное (математическое) описание случайных факторов, подлежащих учету в модели;

• выбор механизма изменения и масштаба модельного времени.

Работа (активность) — это единичное действие системы по обработке (преобразованию) входных данных. В зависимости от природы моделируемой системы под входными данными могут пониматься информационные данные или какие-либо материальные ресурсы. Каждая из работ характеризуется временем выполнения и потребляемыми ресурсами.

Под ***процессом*** понимают логически связанный набор работ. Некоторые процессы могут рассматриваться, в свою очередь, как работы в процессе более высокого уровня. Процесс характеризуется совокупностью статических и динамических характеристик.

К *статическим* характеристикам процесса относятся:

• длительность;

• результат;

• потребляемые ресурсы;

• условия запуска (активизации);

• условия останова (прерывания).

В общем случае статические характеристики процесса не изменяются в ходе его реализации, однако, при необходимости любая из них может быть представлена в модели как случайная величина, распределенная по заданному закону.

Динамической характеристикой процесса является его состояние (активен или находится в состоянии ожидания).

Моделирование в терминах процессов производится в тех случаях, когда система оценивается по каким-либо временным показателям, либо с точки зрения потребляемых ресурсов.

Например, при оценке производительности вычислительной сети обработка заданий может быть представлена в модели как совокупность соответствующих процессов, использующих ресурсы сети (оперативную память, пространство на жестких дисках, процессорное время, принтеры и т. д.).

В том случае, если модель строится с целью изучения причинно-следственных связей, присущих системе, динамику системы целесообразно описывать в терминах событий.

Событие представляет собой мгновенное изменение некоторого элемента системы или состояния системы в целом.

Событие характеризуется:

• условиями (или законом) возникновения;

• типом, который определяет порядок обработки (дисциплину обслуживания) данного события;

• нулевой длительностью.

Обычно события подразделяют на две категории:

события следования, которые управляют инициализацией процессов (или отдельных работ внутри процесса);

события изменения состояний (элементов системы или системы в целом).

Как было отмечено, механизм событий используется в качестве основы построения моделей, предназначенных для исследования причинно-следственных связей в системах при отсутствии временных ограничений. К таким задачам можно отнести, например, некоторые задачи по оценке надежности.

Еще один способ имитационного моделирования систем основан на использовании понятия транзакта.

Транзакт — это некоторое сообщение (заявка, на обслуживание), которое поступает извне на вход системы и подлежит обработке. В некоторых случаях, например, при моделировании автоматизированных систем управления, более удобно проследить функционирование системы именно относительно алгоритма обработки транзакта. В рамках одной ИМ могут рассматриваться транзакты нескольких типов. Каждый транзакт характеризуется соответствующим алгоритмом обработки и необходимыми для его реализации ресурсами системы. Учитывая это, прохождение транзакта по системе можно в некоторых случаях рассматривать как последовательную активизацию процессов, реализующих его обработку («обслуживание заявки»).

В связи с упоминанием термина «обслуживание заявки» уместно вспомнить о существовании теории массового обслуживания. При разработке и исследовании имитационных моделей на основе транзактов целесообразно использовать методику и показатели, применяемые при анализе систем массового обслуживания.

5. РАЗРАБОТКА ИМИТАЦИОННОЙ МОДЕЛИ

5.1. Управление модельным временем

Приступая к изучению механизмов управления модельным временем, уместно поговорить о роли времени в имитационном моделировании. Ранее было отмечено, что имитационное моделирование представляет собой наблюдение за поведением системы в течение некоторого промежутка времени. Конечно, далеко не во всех статистических испытаниях фактор времени играет ведущую роль, а в некоторых и вообще может не рассматриваться. Но значительно больше таких задач, в которых оценка эффективности моделируемой системы напрямую связана с временными характеристиками ее функционирования. К ним относятся задачи по оценке производительности, некоторые задачи по оценке надежности, качества распределения ресурсов, а также все задачи, связанные с исследованием эффективности процессов обслуживания. Характерной особенностью большинства практических задач является то, что скорость протекания рассматриваемых в них процессов значительно ниже скорости реализации модельного эксперимента. Например, если моделируется работа авторемонтной мастерской в течение недели, вряд ли кому-то придет в голову воспроизводить этот процесс в модели в таком же масштабе времени. А в ряде задач требуется именно реализация реального масштаба времени.

При разработке практически любой имитационной модели и планировании проведения модельных экспериментов необходимо соотносить между собой три представления времени:

• реальное время, в котором происходит функционирование имитируемой системы;

• модельное (или, как его еще называют, системное) время, в масштабе которого организуется работа модели;

• машинное время, отражающее затраты времени ЭВМ на проведение имитации.

С помощью механизма модельного времени решаются следующие задачи:

1) отображается переход моделируемой системы из одного состояния в другое;

2) производится синхронизация работы компонент модели;

3) изменяется масштаб времени «жизни» (функционирования) исследуемой системы;

4) производится управление ходом модельного эксперимента.

5) 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 находится, как правило, однопроцессорная вычислительная система, а модель может содержать значительно большее число одновременно работающих подсистем. Поэтому действительно параллельная (одновременная) реализация всех компонент модели невозможна. Даже если используется так называемая распределенная модель, реализуемая на нескольких узлах вычислительной сети, совсем необязательно число узлов будет совпадать с числом одновременно работающих компонент модели. Следует отметить, что реализация квазипараллельной работы компонент модели является достаточно сложной технической задачей. Некоторые возможные методы ее решения рассматриваются в следующем разделе.

Ранее были названы два метода реализации механизма модельного времени — с постоянным шагом и по особым состояниям.

Выбор метода реализации механизма модельного времени зависит от назначения модели, ее сложности, характера исследуемых процессов, требуемой точности результатов и т. д.

При использовании метода ***постоянного шага*** 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 постоянного шага предпочтительнее, если:

• события появляются регулярно, их распределение во времени достаточно равномерно;

• число событий велико и моменты их появления близки;

• невозможно заранее определить моменты появления событий.

Данный метод управления модельным временем достаточно просто реализовать в том случае, когда условия появления событий всех типов в модели можно представить как функцию времени.

Пусть, например, событие состоит в том, что летящий самолет пересекает некоторый воздушный рубеж, расстояние до которого равно *R*. Если самолет движется по прямой с постоянной скоростью *V*, то можно вычислять путь, пройденный самолетом, с интервалом времени Δt: $S = S + V \cdot \Delta t$. Соответственно событие считается наступившим, если выполняется условие $S > R$, а момент времени наступления события принимается равным $n \cdot \Delta t$, где n — номер шага моделирования, на котором условие стало истинным.

Выбор величины шага моделирования является нелегким и очень важным делом. Универсальной методики решения этой проблемы не существует, но во многих случаях можно использовать один из следующих подходов:

• принимать величину шага равной средней интенсивности возникновения событий различных типов;

• выбирать величину Δt равной среднему интервалу между наиболее частыми (или наиболее важными) событиями.

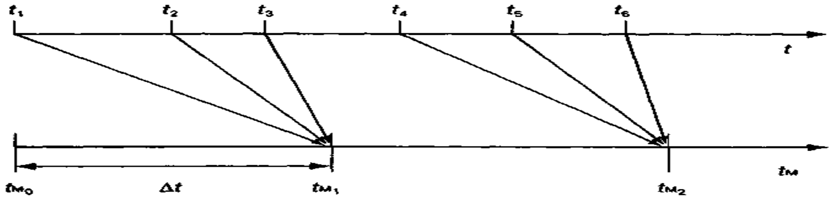


Рис. 5.3

• принимать величину шага равной средней интенсивности возникновения событий различных типов;

• выбирать величину Δt равной среднему интервалу между наиболее частыми (или наиболее важными) событиями.

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.

Метод моделирования по особым состояниям сложнее в реализации, так как для него требуется разработка специальной процедуры планирования событий (так называемого календаря 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 реализовать квазипараллельную обработку 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.

Таким образом:

- Выбор механизма изменения модельного времени определяет и технологию реализации имитационной модели.
- На выбор метода моделирования влияет целый ряд факторов, однако определяющим является тип моделирующей системы: для дискретных систем, события в которых распределены во времени неравномерно, более удобным является изменение модельного времени по особым состояниям.

Если в модели должны быть представлены компоненты реальной системы, работа которых измеряется в разных единицах времени, то они должны быть предварительно приведены к единому масштабу.

5.2. Моделирование параллельных процессов

Практически любая более или менее сложная система имеет в своем составе компоненты, работающие одновременно, или, как принято говорить на языке техники, параллельно. 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.

5.2.1.Виды параллельных процессов в сложных системах

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

Пример асинхронных ПП 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).

Один и тот же процесс может быть синхронным по отношению к одному из активных ПП и асинхронным по отношению к другому. Так, при работе вычислительной сети по технологии «клиент-сервер» каждый из узлов сети синхронизирует свою работу с работой сервера, но не зависит от работы других узлов.

Подчиненный ПП — создается и управляется другим процессом (более высокого уровня). Весьма характерным примером таких процессов является ведение боевых действий подчиненными подразделениями.

Независимый ПП**—** не является подчиненным ни для одного из процессов. Скажем, после запуска неуправляемой зенитной ракеты ее полет можно рассматривать как независимый процесс, одновременно с которым самолет ведет боевые действия другими средствами.

Способ организации параллельных процессов в системе зависит от физической сущности этой системы.

Остановимся несколько подробнее на особенностях реализации параллельных процессов в вычислительных системах (ВС). Это обусловлено следующей причиной.

Разработка и использование любой ИМ предполагает ее программную реализацию и исследование с применением ВС. Поэтому для реализации моделей, имитирующих параллельные процессы, в некоторых случаях применимы механизмы, используемые при выполнении параллельных вычислений.

Вместе с тем, реализация параллельных процессов в ВС имеет свои особенности:

- на уровне задач вычислительные процессы могут быть истинно параллельными только в многопроцессорных ВС или вычислительных сетях;
- многие ПП используют одни и те же ресурсы, поэтому даже асинхронные ПП в пределах одной ВС вынуждены согласовывать свои действия при обращении к общим ресурсам;
- в ВС дополнительно используется еще два вида ПП: родительский и дочерний ПП; особенность их состоит в том, что процесс-родитель не может быть завершен, пока не завершатся все его дочерние процессы.

В силу перечисленных особенностей для организации взаимодействия параллельных процессов в ВС используются три основных подхода:

• на основе «взаимного исключения»;

• на основе синхронизации посредством сигналов;

• на основе обмена информацией (сообщениями).

«Взаимное исключение» предполагает запрет доступа к общим ресурсам (общим данным) для всех ПП, кроме одного, на время его работы с этими ресурсами (данными).

Синхронизация подразумевает обмен сигналами между двумя или более процессами по установленному протоколу. Такой «сигнал» рассматривается как некоторое событие, вызывающее у получившего его процесса соответствующие действия.

Часто возникает необходимость передавать от одного ПП другому более подробную информацию, чем просто «сигнал-событие». В этом случае процессы согласуют свою работу на основе обмена сообщениями.

Перечисленные механизмы реализуются в ВС на двух уровнях — системном и прикладном.

Механизм взаимодействия между ПП на системном уровне определяется еще на этапе разработки ВС и реализуется в основном средствами операционной системы (частично — с использованием аппаратных средств).

На прикладном уровне взаимодействие между ПП реализуется программистом средствами языка, на котором разрабатывается программное обеспечение.

Наибольшими возможностями в этом отношении обладают так называемые языки реального времени (ЯРВ) и языки моделирования.

Языки реального времени — это языки, предназначенные для создания программного обеспечения, работающего в реальном масштабе времени, например для разработки различных автоматизированных систем управления (предприятием, воздушным движением и т. д.). К ним, в частности, относятся: язык *Ада,* язык *Модула* и практически единственный отечественный язык реального времени — *Эль-76* (использовавшийся в многопроцессорных вычислительных комплексах семейства «Эльбрус»).

5.2.2. Методы описания параллельных процессов в системах и языках моделирования

Языки моделирования по сравнению с языками реального времени требуют от разработчика значительно менее высокого уровня подготовки в области программирования, что обусловлено двумя обстоятельствами:

• во-первых, средства моделирования изначально ориентированы на квазипараллельную обработку параллельных процессов;

• во-вторых, механизмы реализации ПП относятся, как правило, к внутренней организации системы (языка) моделирования и их работа скрыта от программиста.

В практике имитационного моделирования одинаково широко используются как процессно-ориентированные языки (системы) моделирования, например *SIMULA,* так и языки, ориентированные на обработку транзактов (например, язык *GPSS).* В тех и других используются аналогичные методы реализации квазипараллелизма, основанные на ведении списков событий. В процессно-ориентированных системах используются списки событий следования, а в транзактных системах — списки событий изменения состояний.

Современные языки и системы моделирования, ориентированные на использование в среде многозадачных операционных систем типа Windows, частично используют их механизмы управления процессами, что делает их применение еще более эффективным. В пакете MATLAB также имеется собственный язык моделирования, и к нему в полной мере можно отнести сказанное выше. Тем не менее во многих случаях оказывается полезным знание общего механизма реализации ПП в языках моделирования.

Рассмотрим его применительно к моделированию на основе траказктов.

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.

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 (PI, P2, P3).*

Временная диаграмма работы системы при обслуживании одного транзакта каждого типа показана на рис.2.7.

На рисунке события, относящиеся к процессу P1, обозначены как $C1_i$, относящиеся к P2 и к P3 — соответственно как $C2_i$. и $C3_i$. Моменты времени $t_{вх}$ и $t_{вых}$ соответствуют началу и окончанию обслуживания транзакта.

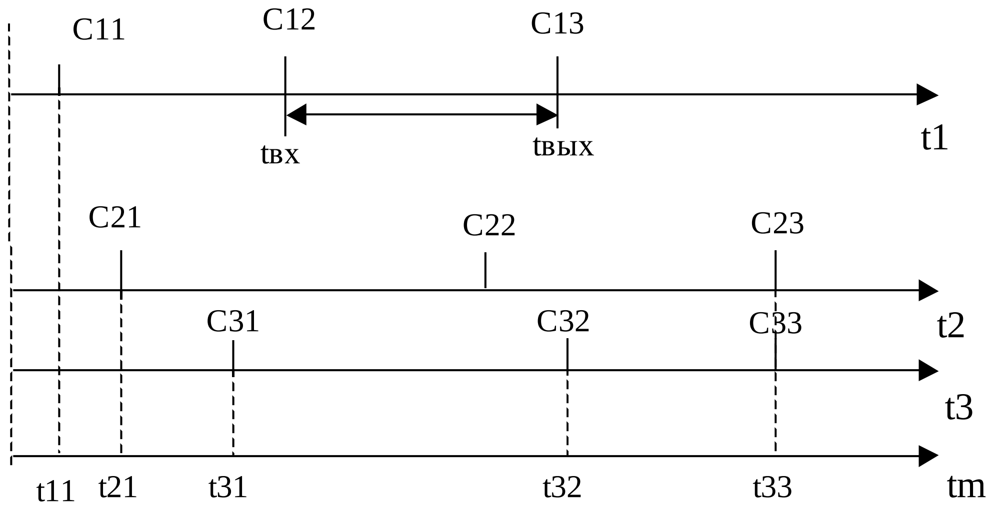


Рис. 5.6

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 tм=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 приведенной ниже таблице.

t	***Список текущих событий***	***Список будущих событий***
0	*0*	*C11, C21,C31*
t11	*C11*	*C21, C31, C12*
t21	*C21*	*C31, C12, C22*
t31	*C31*	*C12, C22, C32*
t12	*C12*	*C22, C32, C13*
t22	*C22*	*C32, C13, C23*
t32	*C32*	*C13, C23, C33*
t13	*C13*	*C23, C33*
t23	*C23, C33*	

◄

Многие авторы книг по имитационному моделированию считают, что знание механизма ведения списков событий просто необходимо разработчику модели; умение проследить в динамике цепь происходящих в модели событий, во-первых, повышает уверенность создателя модели в том, что она работает правильно и, во вторых, существенно облегчает процесс отладки и модификации модели.

5.2.3. Применение сетевых моделей для описания параллельных процессов

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 программу, ее создатель должен воспользоваться каким-то менее формальным и более наглядным средством описания логики работы будущей программы. Это требование не является обязательным, т.к. при наличии достаточного опыта программа не очень сложной модели может быть написана сразу. Однако 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 такие схемы совершенно не приспособлены для описания параллельных процессов.

Одним из наиболее элегантных и весьма распространенных средств описания параллельных процессов — описание ***сетями Петри.*** Рассмотрим те основные сведения, которые необходимы с точки зрения реализации технологии имитационного моделирования параллельных процессов.

Одно из основных достоинств аппарата сетей Петри заключается в том, что они могут быть представлены как в графической форме (что обеспечивает наглядность), так и в аналитической (что позволяет автоматизировать процесс их анализа).

При графической интерпретации сеть Петри представляет собой граф особого вида, состоящий из вершин двух типов — *позиций* и *переходов,* соединенных ориентированными дугами, причем каждая дуга может связывать лишь разнотипные вершины (позицию с переходом или переход с позицией). Вершины-позиции обозначаются кружками, вершины-переходы — черточками. С содержательной точки зрения, переходы соответствуют событиям, присущим исследуемой системе, а позиции — условиям их возникновения. Таким образом, совокупность переходов, позиций и дуг позволяет описать причинно-следственные связи, присущие системе, но в статике. Чтобы сеть Петри «ожила», вводят еще один вид объектов сети — так называемые *фишки,* или метки позиций. Переход считается активным (событие может произойти), если в каждой его входной позиции есть хотя бы одна фишка. Расположение фишек в позициях сети называется ***разметкой сети*** (пример перемещения фишек по сети приведен на рис.5.6).

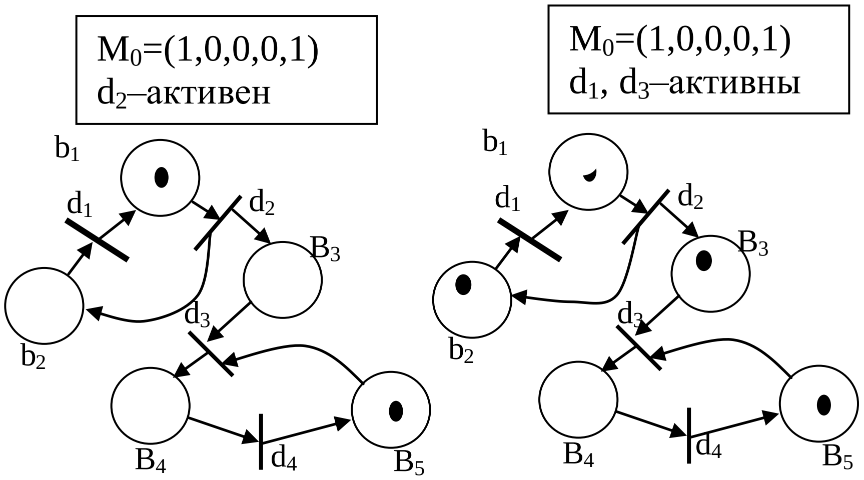


Рис. 5.6

В аналитической форме сеть Петри может быть представлена следующим образом:

$$P=(B,D,I,O,M),$$

где $\boldsymbol{B} = \{b_i\}$ — конечное непустое множество позиций;

$\boldsymbol{D} = \{d_i\}$ — конечное непустое множество переходов;

$\boldsymbol{I : BxD}$ -> *0,1* — входная функция (прямая функция инцидентности), которая для каждого перехода задает множество его входных позиций;

$\boldsymbol{O : DxB}$ -> 0,1 — выходная функция (обратная функция инцидентности), которая для каждого перехода задает множество его входных позиций;

$\boldsymbol{M}$ — функция разметки сети, $\boldsymbol{M : B}$ -> 0, 1, 2,... — ставит каждой позиции сети в соответствие неотрицательное целое число.

С учетом введенных обозначений необходимое условие срабатывания перехода *dj* может быть записано следующим образом:

$$\forall b_i \in I(d_j)\ \{M(b_i) \geq 1\}$$

(*для всех входных позиций разметка должна быть >1*).

Срабатывание перехода d_j изменяет разметку сети *M(B)* на разметку *M'(B)* по следующему правилу:

$$M'(B)=M(B)-I(d_j)+O(d_j),$$

то есть переход d_j изымает по одной метке из каждой своей входной позиции и добавляет по одной метке в каждую из выходных позиций. Смену разметки обозначают так:

$$M_o \overset{d_j}{/\!\!-\!\!-} M'$$

Входная и выходная функции сети Петри ($\boldsymbol{I}$ и $\boldsymbol{O}$) позволяют описать любую сеть с помощью двух матриц размера *m x n* (матриц входных и выходных позиций), имеющих следующую структуру:

	d_1	d_2	...	d_j	...	d_n
b_1	0	1	...	0	...	0

b_2	1	1	...	0	...	1
...	...	...	...	...	...	...
b_j	0	1	...	0	...	1
...	...	...	...	...	...	...
b_m	1	0	...	1	...	0

Основные направления анализа сети Петри следующие:

1. *Проблема достижимости:* в сети Петри с начальной разметкой M_0 требуется определить, достижима ли принципиально некоторая разметка M' из M_0. С точки зрения исследования моделируемой системы, эта проблема интерпретируется как проблема достижимости (реализуемости) некоторого состояния системы.

2*. Свойство живости*. Под живостью перехода понимают возможность его срабатывания в данной сети при начальной разметке M_0. Анализ модели на свойство живости позволяет выявить невозможные состояния в моделируемой системе (например, неисполняемые ветви в программе).

3. *Безопасность сети*. Безопасной является такая сеть Петри, в которой ни при каких условиях не может появиться более одной метки в каждой из позиций. Для исследуемой системы это означает возможность функционирования ее в стационарном режиме. На основе анализа данного свойства могут быть определены требования к буферным накопителям в системе.

Итак, достоинства сетей Петри заключаются в том, что они:

1) позволяют моделировать ПП всех возможных типов с учетом вероятных конфликтов между ними;

2) обладают наглядностью и обеспечивают возможность автоматизированного анализа;

3) позволяют переходить от одного уровня детализации описания системы к другому (за счет раскрытия/закрытия переходов).

Вместе с тем, сети Петри имеют ряд недостатков, ограничивающих их возможности. Основной из них — время срабатывания перехода считается равным 0, что не позволяет исследовать с помощью сетей Петри временные характеристики моделируемых систем.

В результате развития аппарата сетей Петри был разработан ряд расширений. Одно из наиболее мощных — так называемые .Е-сети (evaluation — «вычисления», «оценка») — «оценочные сети».

В отличие от сетей Петри, в Е-сетях:

1) имеются несколько типов вершин-позиций: простые позиции, позиции-очереди, разрешающие позиции;

2) фишки (метки) могут снабжаться набором признаков (атрибутов);

3) с каждым переходом может быть связана ненулевая задержка и функция преобразования атрибутов фишек;

4) введены дополнительные виды вершин-переходов.

5) в любую позицию может входить не более одной дуги и выходить также не более одной.

В связи с этим любой переход может быть описан тройкой параметров:

$$\boldsymbol{d_j=(S,t(d_j),\rho(dj)),}$$

где S — тип перехода,

$t(d_j)$, — функция задержки,

$\rho(d_j)$ — функция преобразования атрибутов.

Особенности Е-сетей существенно расширяют их возможности для моделирования дискретных систем вообще и параллельных процессов в частности. Технология моделирования систем в виде Е-сетей может быть реализована с помощью инструмента SIMULINK, входящего в состав пакета M ATLAB.

6. МОДЕЛИРОВАНИЕ СЛУЧАЙНЫХ ФАКТОРОВ

Имитационная модель позволяет исследовать поведение различных систем с учетом влияния 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).

В основе всех методов и приемов моделирования случайных факторов лежит использование базовой случайной величины -- случайных чисел, имеющих равномерное распределение на интервале [0; 1].

6.1.Построение датчиков БСВ

6.1.1. Датчики БСВ

Базовой случайной величиной (БСВ) в статистическом моделировании называют непрерывную случайную величину Z, равномерно распределенную на интервале (0,1). Ее плотность распределения вероятностей имеет вид:

$$f(z) = \begin{cases} 1, & z \in (0,1]; \\ 0, & z \notin (0,1], \end{cases}$$

Математическое ожидание M[z] и дисперсия D[z] БСВ составляют

$$M[z] = \frac{1}{2},$$

$$D[z] = \frac{1}{12},$$

соответственно.

БСВ моделируется на ЭВМ с помощью *датчиков* БСВ. Датчик БСВ – это устройство или программа, выдающая по запросу одно или несколько независимых значений z_1 , ..., z_n БСВ.

Датчики БСВ могут быть трех типов: табличные, физические и программные.

Табличный датчик БСВ – это просто таблица случайных чисел. Основной недостаток такого датчика – ограниченное количество случайных чисел в таблицах. А в статистическом эксперименте часто требуется не ограниченное заранее их количество.

Физический датчик БСВ – это специальное радиоэлектронное устройство в ЭВМ, содержащее источник электронного шума. Шум преобразуется в случайные числа с распределением. Недостатки физического датчика БСВ: невозможность повторения каких-либо ранее полученных реализаций z_1, ... , z_n без их предварительной записи в память ЭВМ, схемная нестабильность и сложность тиражирования датчика.

Программный датчик БСВ обычно вычисляет значения z_1, z_2,..., по какой-либо рекуррентной формуле типа

$$z_i = f\,(z_n),$$

при заданном стартовом значении z_0.

Заданное значение z_0 полностью определяет всю последовательность реализаций z_1, z_2,..., поэтому z часто называют *псевдослучайной* величиной. Но ее статистические свойства идентичны свойствам "чисто случайной" последовательности, что и обеспечивает успех статистического моделирования.

Программный датчик БСВ имеет следующие преимущества: простота создания датчика, простота применения, простота тиражирования, надежность, быстродействие, высокая точность достижения необходимых статистических свойств, сравнимая с точностью представления вещественных чисел, компактность, повторяемость, когда это нужно, любых последовательностей случайных значений без их предварительного запоминания.

В дальнейшем мы будем рассматривать только программные датчики БСВ.

Имея датчик БСВ Z, можно промоделировать любые случайные факторы: непрерывные или дискретные случайные величины (как простые, так и многомерные), случайные события, случайные процессы и поля и т.д. Для этого достаточно соответствующим образом преобразовать последовательность z_1, z_2, ... Поэтому БСВ Z и называют базовой.

Теоретически в качестве базовой можно было бы взять почти любую случайную величину. Использование СВ z с распределением обусловлено технологическими соображениями: простотой и экономичностью датчика, простотой преобразования Z в другие случайные факторы, относительной простотой тестирования датчика.

6.1.2.Метод середины квадрата

Метод середины квадрата предложен для получения псевдослучайных чисел Д. фон Нейманом в 1946 г. Один из вариантов этого метода заключается в следующем.

1. Возьмем произвольное n-разрядное число.
2. Возведем полученное число в квадрат и, если необходимо, добавим к результату слева нули до $2n$-разрядного числа.
3. Возьмем четыре цифры из середины $2n$-разрядного в качестве нового случайного n-разрядного числа.
4. Если нужны еще случайные числа, то перейдем к пункту 2.

Например, если взять в качестве начального числа 1994, то из него получается следующая последовательность псевдослучайных чисел: 9760

2576 6357 4114 9249 5440 5936 2360 5696 4444 7491 1150 3225 4006 0480 2304 3084 5110 1121 2566 ...

Сам по себе метод середины квадрата не получил широкого распространения, так как выдает "больше чем нужно малых значений". Но открытый в нем принцип используется во многих, если не во всех, более поздних датчиках БСВ. Этот принцип состоит в вырезании нескольких цифр из результата какой-либо операции над числами.

6.1.3. Мультипликативный конгруэнтный метод

Так называемый мультипликативный конгруэнтный датчик БСВ задается двумя параметрами: модулем m и множителем k. Обычно это достаточно большие целые числа.

При заданных m, k числа z_1, z_2, ..., вычисляются по рекуррентной формуле:

$$A_i = (kA_{i-1}) \bmod m, \quad i = 1, 2, ..., \tag{6.1}$$

$$z_i = A_i / m,$$

где m – модуль,

k – множитель,

A_0 – начальное значение,

mod – операция вычисления остатка от деления kA_{i-1} на m.

Таким образом, A_1 определяется как остаток от деления kA_0 на m; A_2 - как остаток от деления kA_1 на m и т.д. Поскольку все числа A_i – это остатки от деления на m, то $0 \leq A_i < m$. Разделив последнее неравенство на m, видим, что $0 \leq A_i / m < 1$, т. е. $0 \leq z_i < 1$.

Из неравенства $0 \leq A_i < m$ вытекает также, что датчик (6.1) дает периодическую последовательность A_i. Действительно, число всех возможных остатков от 0 до m - 1 равно m и, рано или поздно, на каком-то шаге i обязательно появится значение A_i, уже встречавшееся ранее. С этого момента последовательность A_i “зациклится".

Длина периода T будет не больше m - 1. Например, если встретится остаток $A_i = 0$, то далее, согласно (4.1), будет $A_{i+1} = 0$, $A_{i+2} = 0$, ... , т.е. длина периода $T = 1$. Ненулевых же остатков в интервале $0 \leq A_i < m$ всего m - 1, и, если все они войдут в период, будет $T = m - 1$. Это имеет место, например, при $m = 13$, $k = 7$; в этом случае ряд A_i выглядит так:

$$\underbrace{1, 7, 10, 5, 9, 11, 12, 6, 3, 8, 4, 2,}_{T = m - 1 = 12}\ 1, 7, ...$$

Поскольку в качестве случайной можно использовать лишь подпоследовательность A_i внутри одного периода, то параметры датчика выбирают так, чтобы длина периода T была максимальной. С учетом ограничения $T \leq m$ - 1 модуль m берут максимально возможным. Чтобы упростить вычисление остатков по (6.1), для двоичных ЭВМ часто берут $m = 2^n$. Рекомендуется также брать достаточно большой множитель k, причем взаимно простой с m.

В можно найти подробные рекомендации по выбору параметров m, k и начального значения A_0 . Заметим, однако, что в настоящее время не известны правила, которые гарантировали бы высокое качество датчика без его специального статистического тестирования.

Датчик (4.1) называют мультипликативно-конгруэнтным потому, что он использует две основные операции – умножение (англ. multiplication) и вычисление остатка (в теории чисел – получение конгруэнтного числа). Можно было бы поэтому перевести его название и как "множительно-остатковый датчик".

Обратим внимание также и на то, что операция вычисления остатка воплощает здесь упоминавшийся в п. 4.1.2 неймановский принцип вытаскивания цифр. Это становится очевидным, если записывать числа в системе счисления с основанием m. Тогда операция $X \bmod m$ означает выбор последней цифры из числа X. Для $m = 2^n$ операция $X \bmod m$ означает также выделение последних n цифр из двоичной записи числа X.

В качестве примера рассмотрим таблицу параметров датчиков, предлагаемых в некоторых публикациях и программных продуктах.

Место использования датчика (программный продукт или публикация)	модуль m	множитель k
Язык моделирования СИМУЛА	2^{35}	5^{16}
Пакеты LLRANDOM, IMSL	$2^{31} - 1$ (простое число)	16807
Язык моделирования SIMSCRIPT	$2^{31} - 1$	63036001

6.2. Характеристики датчиков базовых случайных величин

Практика показывает, что результаты имитационного моделирования существенно зависят от качества используемых последовательностей псевдослучайных чисел. Поэтому используемые в имитационной модели генераторы случайных чисел должны пройти тесты на пригодность. Основные анализируемые характеристики генерируемых датчиком последовательностей:

- равномерность;
- стохастичность (случайность);
- независимость.

Рассмотрим методы проведения такого анализа, наиболее часто применяемые на практике.

6.2.1. Тестирование равномерности

Обозначим равномерное распределение вероятностей на интервале (0,1) через $R[0,1]$. Тогда утверждение, что БСВ Z имеет распределение $R[0,1]$, можно кратко записать в виде $z \sim R[0,1]$.

С помощью статистических тестов проверяют два свойства датчика, делающих его точной моделью идеальной БСВ, – это *равномерность* распределения чисел Z_i, выдаваемых датчиком на интервале (0,1), и их статистическая *независимость*. При этом числа z_i рассматривают как реализации некоторой **СВ.**, т.е. как статистическую выборку.

Достаточно простым методом проверки равномерности распределения является частотный тест. Он основан на законе больших чисел и выполняется по следующему алгоритму.

1. Разобьем интервал (0,1) на K равных отрезков (например, $K = 10$).
2. Сгенерируем n чисел $z_1,..., z_n$ с помощью тестируемого датчика БСВ (например, $n = 100$).
3. Подсчитаем, сколько чисел попало в каждый из k отрезков, т.е. найдем числа попаданий $n_1,...,n_k$.
5. Рассчитаем относительные частоты попаданий в отрезки:

$$\hat{p} = \frac{n_1}{n}, ..., \hat{p} = \frac{n_k}{n}$$

5. Построим гистограмму частот $\hat{p}_1, ..., \hat{p}_k$ на K отрезках интервала (0,1).
6. Повторим действия (2) – (5) для большего значения n (например, для n =10 000).
7. Оценим по полученным гистограммам сходимость каждой частоты $\hat{p}_i$ к вероятности $p = 1/K$ того, что БСВ попадет в i-й отрезок. Согласно закону больших чисел должно быть

$$\hat{p}_i \xrightarrow{p} \frac{1}{K}, \qquad n \to \infty \tag{6.2.}$$

Это значит, что высоты столбиков во второй гистограмме должны в целом быть ближе к уровню $1/K$, чем в первой.

Тестирование датчика на равномерность можно совместить с оцениванием **математического ожидания *m**** и дисперсии S^2. Оценки ***m**** и S^2 рассчитываются соответственно по формулам:

$$m^* = \frac{1}{n}\sum_{i=1}^{n} z_i \tag{6.3}$$

$$S^2 = \frac{1}{n-1}\sum_{i=1}^{n}(z_i - m^*)^2 \tag{6.4}$$

С ростом n оценки $\hat{M}$ и $\hat{D}$ должны сходиться по вероятности к точным значениям $M(z) = 1/2$, $D(z) = 1/12 = 0.08333...$.

6.2.2. Тестирование стохастичности

Рассмотрим один из основных методов проверки – метод комбинаций. Суть его сводится к следующему. Выбирают достаточно большую последовательность случайных чисел x_i и для нее определяют вероятность появления в каждом из x_i ровно j единиц. При этом могут анализироваться как все разряды числа, так и только l старших. Теоретически закон появления j единиц в l разрядах двоичного числа может быть описан как биномиальный закон распределения (исходя из независимости отдельных разрядов).

Тогда при длине выборки N ожидаемое число появлений случайных чисел x_i с j единицами в проверяемых l:

$$n_j = N \cdot C_l^j p_1^j p_0^{l-j} = 2^{-l} N \cdot C_l^j$$

Для полученной последовательности определяется эта же характеристика. Проверка соответствия реального значения теоретическому выполняется с помощью одного из статистических критериев согласия.

6.2.3. Тестирование независимости

Простейшую проверку статистической независимости реализаций z_1, z_2, ..., можно осуществить, оценивая корреляцию между числами z_i и z_{i+s}, отстоящими друг от друга на шаг $s > 1$.

Для вывода формулы, по которой можно рассчитать коэффициент корреляции чисел z_i и z_{i+s}, рассмотрим две произвольные **с.в.** x, y. Коэффициент корреляции определяется для них формулой:

$$R(x,y) = \frac{M(xy) - M(x)M(y)}{\sqrt{D(x)D(y)}} \tag{6.5}$$

С ростом n оценка R' должна приближаться к нулю, в противном случае датчик БСВ не отвечает требованию независимости.

Конечно, если R' сходится к нулю, то это еще не гарантирует наличие независимости, но все же один из тестов оказывается успешно выдержанным. При желании всегда можно продолжить испытания датчика другими методами.

Еще одна важная характеристика датчика СЧ — **длина отрезка периодичности *L***. Если в основу работы датчика положен мультипликативный метод, то оценить L несложно: она определяется величиной константы *M*.

6.3. Случайные события и их имитация

6.3.1.Имитация случайного события

Пусть некоторое событие А происходит с вероятностью P_A. Требуется воспроизвести факт наступления события А. Поставим в соответствие событию А событие В, состоящее в том, что x меньше либо равно P_A, где х здесь и в дальнейшем – случайное число (СЧ) с равномерным на интервале (0,1) законом распределения. Вычислим вероятность события В:

$$P(B) = \int_0^P 1dy = P_A$$

Таким образом, события А и В являются равновероятными. Отсюда следует процедура имитации факта появления события А. Она сводится к проверке неравенства X_A меньше, либо равно Р, а алгоритм заключается в следующем:

1. С помощью датчика случайных чисел (СЧ) получают СЧ X;
2. Проверяют выполнение неравенства Х меньше, либо равно P_A;
3. Если оно выполняется, то событие А – произошло, если нет – то произошло $\bar{A}$

6.3.2. Имитация сложного события

Имитация сложного события, состоящего, например, из двух независимых элементарных событий А и В, заключается в проверке неравенств:

$$\left.\begin{matrix} x_1 \le P_A \\ x_2 \le P_B \end{matrix}\right\},$$

где P_A и P_B – вероятности событий А и В, а x_1 и x_2 – СЧ с равномерным законом распределения.

В зависимости от исхода проверки неравенств (аналогично алгоритму 4.2.1.) делается вывод какой из вариантов:

$AB, A\bar{B}, \bar{A}B, \bar{A}\bar{B}$ имеет место.

6.3.3. Имитация сложного события, состоящего из зависимых событий.

В случае, когда сложное событие состоит из элементарных зависимых событий А и В имитация сложного события производится с помощью проверки следующих неравенств:

$$\left.\begin{matrix} x_1 \le P_A \\ x_2 \le P_{B/A} \end{matrix}\right\} \quad \left.\begin{matrix} x_1 > P_A \\ x_2 \le P_{B/\bar{A}} \end{matrix}\right\} \quad \left.\begin{matrix} x_1 \le P_A \\ x_2 > P_{B/A} \end{matrix}\right\} \quad \left.\begin{matrix} x_1 > P_A \\ x_2 > P_{B/\bar{A}} \end{matrix}\right\}$$

В зависимости от того, какая из этих четырех систем неравенств выполняется, делается вывод о том, какой из этих четырех возможных исходов $AB, A\bar{B}, \bar{A}B, \bar{A}\bar{B}$ имеет место.

В качестве исходных данных задаются P_A, P_B и условная вероятность $P_{B/A}$, вероятность $P_{B/\bar{A}}$ может быть вычислена. По формуле полной вероятности:

$$P(A) = P(A/B)\cdot P(B) + P(A/\overline{B})\cdot P(\overline{B}),$$

где

$P(\overline{B}) = 1 - P(B)$, отсюда легко выразить $P(A/\overline{B})$

6.3.4. Имитация событий, составляющих полную группу

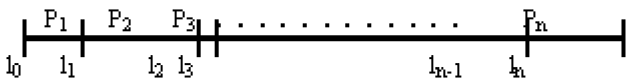
Пусть событие A_i (i=1,n) составляют полную группу, тогда их вероятности P_i, таковы что:

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

Имитация факта появления одного из событий A_i (i=1,n) сводится к проверке следующих неравенств:

$$\sum_{i=0}^{K-1} P_i \le x < \sum_{i=0}^{K} P_i, \qquad K = \overline{1,n}, \qquad P_0 = 0$$

Выполнение К-го неравенства эквивалентно выполнению события A_K. Описанный алгоритм называют иногда алгоритмом “розыгрыша по жребию”. Его можно интерпретировать как установление номера К-го отрезка длинной P_K, на который пало СЧ x, при условии разбиения отрезка единичной длины на отрезки с длинами $P_1, P_2, \ldots P_n$ (рис 4.3.)



$$l_k = \sum_{i=0}^{K} P_i$$

Рис. 4.3

6.4. Имитация непрерывных случайных величин

6.4.1. Метод обратной функции

Пусть непрерывная случайная величина Y задана своим законом распределения:

$$F_\eta(y) = \int_{-\infty}^{y} f_\eta(y)dy,$$

где *f(y)*– плотность распределения вероятностей, а *F(y)*- функция распределения вероятностей. Доказано, что случайная величина

$$\xi = \int_{-\infty}^{\eta} f(y)dy = F(y)$$

распределена равномерно на интервале (0,1).

Отсюда следует, что искомое значение у может быть определено из уравнения:

$$x = \int_{-\infty}^{y} f\ (y)dy \qquad (6.8)$$

которое эквивалентно уравнению:

$$x = f\ (y) \qquad (6.9)$$

где y – значение случайной величины Y, а x – значение СВ X. Решение уравнения (4.9) можно записать в общем виде через обратную функцию

$$y = F\ (x)$$

Основной недостаток метода заключается в том, что интеграл (4.8) не всегда является берущимся, а уравнение (4.9) не всегда решается аналитическими методами.

6.4.2. Метод Неймана (режекции)

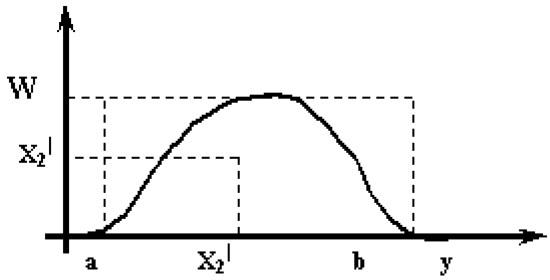
Метод Неймана, так же как метод обратной функции, является методом, позволяющим получить значения СВ в соответствии с заданным законом распределения. Этот метод является достаточно универсальным он применим для моделирования всех СВ, значения которых не выходят за пределы ограниченного интервала (a,b), а также для СВ, законы распределения которых можно аппроксимировать усеченными.

Метод Неймана состоит в следующем:

С помощью датчика случайных чисел получают пару чисел, распределенных равномерно на (0,1) x_1 и x_2.

Путем преобразований (по методу обратной функции получают два числа x_1* и x_2*, равномерно распределенных соответственно на интервалах (a,b) и (o,w), то есть

$x_1^* = a + x_1(b-a)$ и $x_2^* = x_2' W$, где $W = \max f_n(y)$



Из точек с координатами x_1* и x_2* выбирают те, которые попали “под колокол” функции *f(x)*, то есть те точки, для которых $f(x_1^*)<x_2^*$.

Если условие выполнено, то искомое значение y полагают равным x_1*.

6.4.3. Алгоритм получения значения нормально распределенной случайной величины.

Нормальное распределение является наиболее часто встречающимся. Функция плотности распределения вероятностей для него имеет вид:

$$f(y)=\frac{1}{\sqrt{2\pi\sigma}}\exp(-\frac{(y-m)^2}{2\sigma^2})$$

где m – математическое ожидание, а σ^2– дисперсия. Согласно центральной предельной теореме теории вероятностей

$$X=\sum_{i=1}^{k}\xi_i$$

распределена асимптотически нормально, если ξ_i распределены одинаково.

Для практического получения значений X в качестве и ξ_i выбирают равномерно распределенные случайные величины. При этом наиболее часто используют преобразование

$$y=\left(\frac{1}{\sqrt{k/12}}\right)\sum_{i=1}^{k}x_i-\frac{k}{2} \quad (4.10)$$

где x_i – равномерно распределенные на (0,1) случайные числа. При к=12 формула приобретает вид наиболее удобной для расчетов, но она дает достаточно точные результаты уже для k=3,4. Формула (4.10) верна для центрированной (m=0) и нормированной (σ =1) случайной величины.

Для получения y*, распределенного нормально с произвольными m и σ, пользуются дополнительно преобразованием

$$y^*=m+\sigma y \quad (4.11)$$

6.5. Алгоритмы получения значений систем случайных величин (случайных векторов).

6.5.1. Метод аналитических преобразований.

Пусть системы непрерывных случайных величин (x_1, x_2, ..., x_n) задана условными законами распределения x_i (i=1,n). По теореме умножения плотностей распределения: совместная функция плотности распределения вероятностей

$$f(x_1, x_2, \dots x_n)=f_1(x_1)\, f_2(x_2|x_1)\, f_3(x_3|x_1x_2) \dots f_1(x_n|\, x_1,x_2, \dots, x_{n-1}).$$

Для системы двух случайных величин (x_1,x_2), алгоритм получения вектора ее значений сводится к следующему:

Вычисление частной функции плотности для x_1:

$$f_1(x_1)=\int_{-\infty}^{+\infty} f(x_1,x_2)dx_2$$

Получение значения X_1 в соответствии с $f_1(x_1)$ согласно любому методу, например, одному из описанных в предыдущем разделе.

Вычисление частной функции плотности для второй компоненты x_2 системы. Она может быть получена на основании теоремы умножения законов распределения:

$$f_2(x_2/x_1)=\frac{f(x_1,x_2)}{f_1(x_1)}$$

Получение x_2 – значения СВ X_2 любым известным методом в соответствии с найденным законом ее распределения.

Алгоритм может быть обобщен для любого *n*. Однако, практические работы, выполняемые по этому методу, связаны с большими вычислительными трудностями, за исключением тех редких случаев, когда интегралы берутся. Поэтому разработаны другие методы, позволяющие решать задачу получения значений системы непрерывных случайных величин.

6.5.2. Метод разложения по координатным случайным величинам.

Пусть СНСВ задана в рамках теории корреляций: математическими ожиданиями компонент (m_1, m_2, . . . m_n) и матрицей корреляционных моментов:

$$k=\begin{bmatrix} k_{11} & k_{12} & \dots & k_{1n} \\ k_{21} & k_{22} & \dots & k_{2n} \\ \dots & \dots & \dots & \dots \\ k_{n1} & k_{n2} & \dots & k_{nn} \end{bmatrix},\ k_{ij}=k_{ij}, k_{ii}=D(X_i)$$

Доказано, что $x_i,\left(i=\overline{1,n}\right)$ можно получить с помощью их разложения по координатам СВ x_i:

$$\begin{aligned} x_1 &= c_{11}x_1+m_1 \\ x_2 &= c_{12}x_1+c_{22}x_2+m_2 \\ &\dots\dots\dots\dots\dots\dots \\ x_n &= c_{1n}x_1+c_{2n}x_2+\dots+m_2 \end{aligned} \qquad (4.12)$$

где x_i - некоррелированные, центрированные, нормированные нормально распределенные СВ.

Коэффициенты $c_{ij},(i,j=\overline{1,n}, i\le j)$ могут быть достаточно просто получены решением системы уравнений:

$$k_{ij}=c_{1i}c_{1j}+c_{2i}c_{2j}+\dots+c_{ii}c_{jj},(i\le j; i,j=\overline{1,n}) \qquad (4.13)$$

Алгоритм получения значений СНСВ сводится к следующему:

- Решение системы нелинейных уравнений (4.13).
- Получение *n* значений y_i нормированных, центрированных СВ, распределенных нормально.
- Вычисление x_i i=(1,…,n) значений СВ, образующих систему непрерывных случайных величин в соответствии с (4.12).

6.5.3. Алгоритм получения значений системы дискретных случайных величин

Дискретный двумерный вектор СДСВ задается двумерным законом распределения, т.е.

а) матрицей вероятностей $\left\| P_{ij} \right\|, i = \overline{1, n},\ j = \overline{1, m}$, где P_{ij} – вероятность совместного появления i-ого и j-ого значений соответственной первой и второй компоненты, причем:

$$\sum_{i=1}^{n} \sum_{j=1}^{m} P_{ij} = 1.$$

б) двумя векторами возможных значений первой и второй компоненты $\{A_i\}$, $\{B_i\}$, $i = \overline{1, n},\ j = \overline{1, m}$.

Получение значений двумерной дискретной системы случайных величин может осуществляться по следующему алгоритму.

Вычисляют суммы $q_i = \sum_{j=1}^{m} P_{ij}$, $l_k = \sum_{i=1}^{k} q_i$, $k = \overline{1, n}$.

Если X - равномерно распределенное случайное число из интервала (0,1) такое, что $l_{k-1} < x \le l_k$, то считают, что x_1 компонента двумерной дискретной случайной величины получила k-ое значение.

Выбирают k-ую строку $\left\| P_{ij} \right\|$, вычисляют $r_S = \sum_{j=1}^{s} P_{kj}$.

Если вновь полученное с помощью датчика случайных чисел X такое, что вторая компонента получила S-е значение.

Замечание: В алгоритме используется правило “розыгрыша по жребию”, однако надо иметь в виду, что $r_S \ne 1$.